

Grado Universitario en Electrónica Industrial y
Automática
(2016-2017)

Trabajo Fin de Grado

“Implementación de algoritmo de modelado de carriles en la arquitectura de software del Vehículo Inteligente IVVI 2.0”

Pablo Rueda Bravo

Tutor

Carlos Guindel Gomez

Madrid, a 26 de septiembre de 2017



[Incluir en el caso del interés de su publicación en el archivo abierto]

Esta obra se encuentra sujeta a la licencia Creative Commons

Reconocimiento – No Comercial – Sin Obra Derivada

ÍNDICE:

CAPÍTULO 1: INTRODUCCIÓN	5
1.1. HISTORIA DEL COCHE AUTÓNOMO.....	9
1.2. OBJETIVOS	11
1.2.1. IVVI 2.0	11
1.3. ESTRUCTURA DEL CONTENIDO	13
CAPÍTULO 2: ESTADO DEL ARTE	15
2.1. SISTEMAS DE AYUDA A LA CONDUCCIÓN	16
2.1.1. Sistema de visión nocturna de Mercedes:	16
2.1.2. Control de crucero adaptativo (ACC) de Bosch:	17
2.1.3. Detector de fatiga del conductor:	18
2.1.4. Sistema de reconocimiento de señales:	19
2.2. AVISO O ASISTENCIA DE MANTENIMIENTO DE CARRIL:	22
CAPÍTULO 3: RECURSOS UTILIZADOS	27
3.1. ROS	28
3.1.1. INTRODUCCIÓN:	28
3.1.2. CONCEPTOS:	28
3.2. OPENCV	31
3.3. POINT CLOUD LIBRARY (PCL)	32
3.4. GIT.....	33
CAPÍTULO 4: DESCRIPCIÓN GENERAL DEL SISTEMA.....	35
4.1. IMPLEMENTACIÓN DEL ALGORITMO DE DETECCIÓN DE CARRIL	36
4.1.1. DISPARIDAD:.....	37
4.1.2. NUBE DE PUNTOS:	39

4.1.3. EXTRACCIÓN DEL PLANO Y OBTENCIÓN DE PARÁMETROS	
EXTRÍNSECOS:.....	41
4.1.4. BIRD VIEW (Vista de pájaro):	43
4.1.5. LANE DETECTION:.....	44
CAPÍTULO 5: INTEGRACIÓN Y ACTUALIZACIÓN DE LOS NODOS	52
CAPÍTULO 6: RESULTADOS OBTENIDOS	58
CAPÍTULO 7: MARCO REGULADOR.....	64
CAPÍTULO 8: ENTORNO SOCIOECONÓMICO	67
8.1. PRESUPUESTO	68
8.2. IMPACTO SOCIOECONÓMICO:.....	69
CAPÍTULO 9: CONCLUSIÓN Y TRABAJOS FUTUROS	71
9.1. CONCLUSIONES.....	72
9.2. TRABAJOS FUTUROS.....	72
CAPÍTULO 10: BIBLIOGRAFÍA.....	74

ÍNDICE DE FIGURAS

Figura 1: Evolución del número de vehículos registrados en España desde 1990 hasta 2015.....	6
Figura 2: Evolución del número de fallecidos anuales en accidentes de tráfico desde 1960 hasta 2015.....	8
Figura 3: Tesla Model S, 2017	10
Figura 4: Vista exterior del IVVI 2.0 desarrollado por la UC3M	11
Figura 5: Sistema estéreo y cámara infrarroja utilizados para monitorizar la carretera.....	12
Figura 6: Visualización del Sistema de Visión Nocturna de Mercedes en el salpicadero.	17
Figura 7: Representación del Control de Crucero Adaptativo	18
Figura 8: Ejemplo de un Sistema de Detección de fatiga.	19
Figura 9: Sistema de Detección de Señales Verticales de Volvo.....	19
Figura 10: Sistema de detección de peatones de Volvo, 2017	21
Figura 11: Ejemplo de Sistema de aviso de salida involuntaria de carril mediante la colocación de sensores en el parachoques delantero.....	23
Figura 12: Sistema de detección de carril con control sobre la dirección de Audi	24
Figura 13: Vehículo utilizado durante el proyecto ALVINN.	26
Figura 14: Esquema conexión entre nodos, tópicos y servicios.....	31
Figura 15: Esquema general del funcionamiento de secciones principales de Git y sus estados	34
Figura 16: Esquema general nodos	36
Figura 17: Configuración de las cámaras del par estéreo. (b) Relación entre los parámetros para obtener la profundidad Z a partir de la disparidad d.....	38
Figura 18: Comparación de la PC con la imagen original	40
Figura 19: Ejemplo evolución uniformidad de la PC. (a) PC sin método Voxel; (b) PC con tamaño de cubos Voxel medio; (c) PC con tamaño de cubos Voxel alto.....	41

CAPÍTULO 1: INTRODUCCIÓN

Desde la invención del primer automóvil hasta nuestros días, el coche es una de las máquinas que más ha evolucionado, ganando en seguridad, confort, eficiencia y contaminación. Esta evolución ha estado unida al crecimiento sostenido de vehículos que circulan por las vías mundiales. Este incremento está motivado tanto por el crecimiento constante de la población, como al crecimiento económico del sector automovilístico. Por ejemplo, en España, la cadena de valor del automóvil ocupa el 9% de la población activa [1] y es el segundo país fabricante de vehículos de Europa y el octavo del mundo, y en 2016 la producción de vehículos en España fue de 2,885 millones de vehículos, según anunció el Ministerio de Economía. Estas cifras reflejan el incremento que ha sufrido la producción de automóviles en los últimos años, tal y como se refleja en la Figura 1.

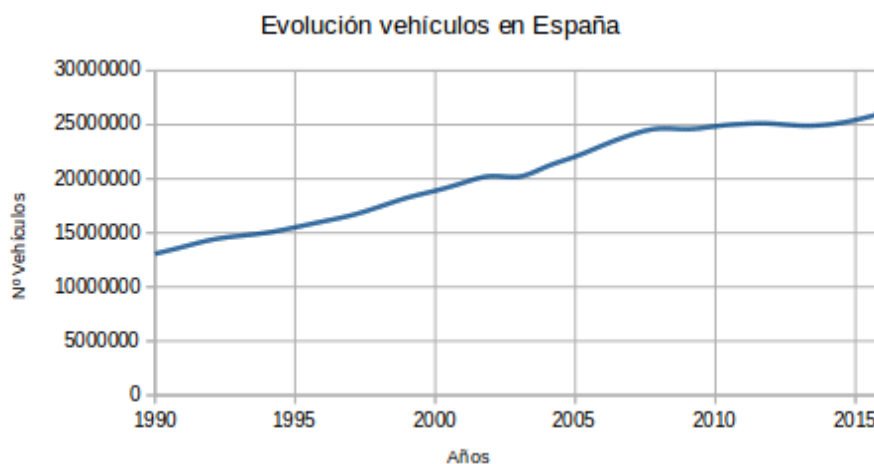


Figura 1: Evolución del número de vehículos registrados en España desde 1990 hasta 2015

A este crecimiento de vehículos en circulación ha ido unido un aumento en el número de accidentes anuales. Los principales factores, según la Dirección General de Tráfico (DGT), [2], causantes de estos accidentes son:

* Exceso de velocidad: La velocidad inadecuada estuvo presente en el 9% de los accidentes con víctimas. En el caso especial de accidentes con víctimas mortales, este porcentaje asciende al 20%.

* Consumo de sustancias psicoactivas: El 43% de los conductores fallecidos presentaron alguna sustancia psicoactiva, siendo alcohol en el 29,3%, drogas de comercio ilegal en el 13,6% y psicofármacos en el 11,4%, según el informe del Instituto Nacional de Toxicología y Ciencias Forenses.

* Distracciones: La distracción aparece en un 29% de los accidentes con víctimas, siendo esta proporción del 36% en las vías interurbanas y 25% en las urbanas.

* Infracciones: En el año 2015, al menos el 45% de los conductores implicados en accidentes ocurridos en vía interurbana había cometido alguna infracción.

Es por estos factores que surgen iniciativas para desarrollar tecnologías que reduzcan los factores humanos en los accidentes. De esta manera, surgen los sistemas de seguridad basados en sensores e imágenes, los cuales actualmente reciben el nombre de ADAS, Sistemas Avanzados de Ayuda a la Conducción. Estos sistemas de ayuda a la conducción están ganando mucha fuerza en la actualidad, ya que estos sistemas reducen de manera considerable accidentes tales como el atropello de peatones, colisión contra otro coche por un frenazo brusco, o colisión contra un obstáculo. También han ganado mucha fuerza en la automoción actual ya que no solo reducen el número de accidentes debidos al factor humano, sino que también aportan comodidad y confort al usuario. Algunos ejemplos de estos sistemas son la ayuda al aparcamiento, o detección de señales verticales de tráfico.

Observando la Figura 2, en la que se representa la evolución de las cifras de fallecidos por accidentes de tráfico con víctimas, se ve como a partir del año 1989 se empiezan a reducir considerablemente el número de fallecidos anuales. Este decrecimiento se debe a la implementación y mejora de sistemas de seguridad tanto para evitar que se produzcan accidentes, como a reducir la gravedad de los daños una vez se produzca.

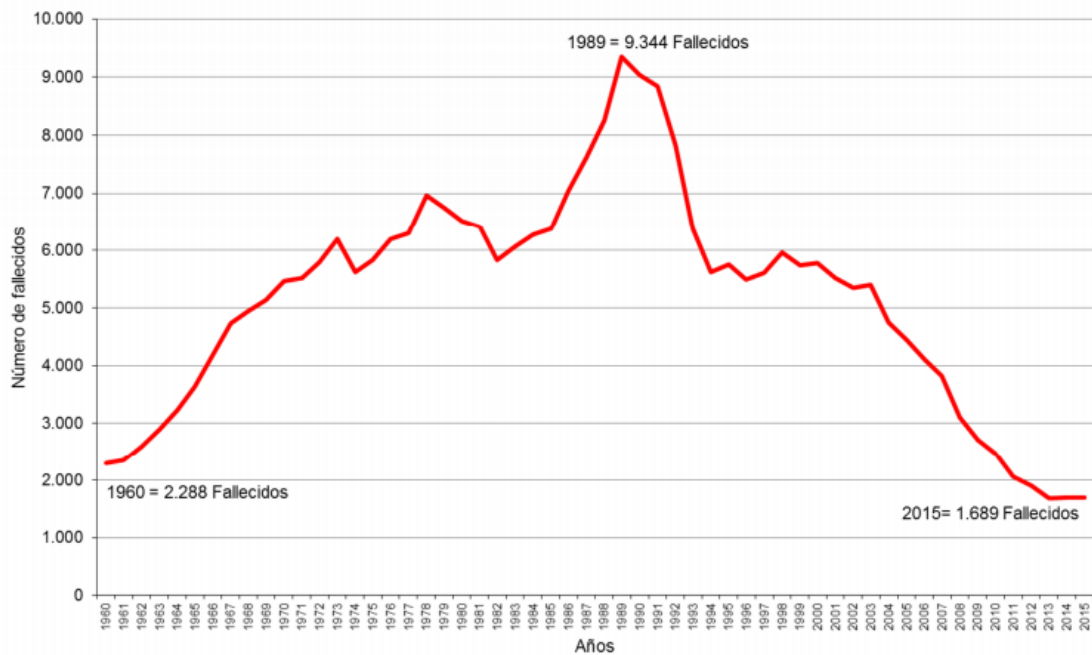


Figura 2: Evolución del número de fallecidos anuales en accidentes de tráfico desde 1960 hasta 2015

En este trabajo se hablará del sistema de detección de carriles, para posteriormente poder realizar acciones tales como la advertencia por salida involuntaria de carril, asistente para mantenimiento de carril, o asistente para cambio de carril. La intención del sistema de detección de carril es la de ser capaz de reconocer la posición del carril en la calzada, así como saber de qué tipo de carril se trata. Con esta información se puede conseguir, a través de actuadores en el coche, controlar el vehículo para poder guiarlo por el carril, tomar decisiones para adelantar a otros coches, tomar desvíos o ceder la incorporación a otros coches. Este sistema está, sobre todo, pensado para el guiado del vehículo en largas distancias, donde el conductor es más propenso a salidas del carril motivadas por despistes, somnolencia, o monotonía.

Otra forma de reducir el factor humano en la circulación es directamente la eliminación de este factor, y así surge la idea de crear el coche autónomo o no tripulado.

1.1. HISTORIA DEL COCHE AUTÓNOMO

El coche autónomo, capaz de desplazarse por sí solo sin necesidad de la acción humana ha sido más una fantasía de ciencia ficción que una realidad, sin embargo, en los últimos años, los avances conseguidos tanto en la visión por computador como en el uso de sensores han convertido esta fantasía en una más que plausible realidad de cara a un futuro no muy lejano. Si bien es cierto que ya existen precedentes de coches autónomos, algunos anteriores a lo que es común pensar.

Tal y como se explica en [3] y [4] la primera aparición de indicios de un coche tripulado sin presencia de conductor se remonta a la década de los años 20 cuando una pequeña empresa de Milwaukee realizó una demostración de manejo de un coche sin conductor, únicamente controlado por radio. No fue hasta 1939 que aparecería de nuevo la idea de coche autónomo, cuando Norman Bel Geddes en la exposición Futurama, en la que se trataba de adivinar como sería la vida 20 años más adelante, presentó un vehículo eléctrico que era controlado por un circuito eléctrico embebido en el pavimento de la carretera. Sin embargo, Ernst Dickmanns y su equipo fueron muy importantes en la historia del vehículo no tripulado, ya que construyeron el primer vehículo robotizado, el cual disponía de visión sacádica, cálculos probabilísticos y computación paralela.

En 1995, un Mercedes-Benz Clase S modificado por el equipo de Dickmanns consiguió realizar un viaje entre Munich y Copenhague y vuelta. Alcanzo velocidades superiores a 175 km/h y con un porcentaje de conducción autónoma del 95%.

A partir del año 2000 y hasta el presente, se inicia un periodo tecnológico marcado por el desarrollo de los ADAS, siglas de *Advanced Driver Assistance Systems*. En este periodo aparecen por primera vez sistemas comerciales tales



Figura 3: Tesla Model S, 2017

como el sistema de asistencia para aparcado inteligente del Toyota Prius en 2003, la alerta por salida de carril del Citroën C4 en 2005, el sistema de visión periférica de Infiniti en 2007, la alerta por somnolencia en 2010, o el asistente para mantenimiento de carril en 2014.

De esta manera, en 2015, una flota de 25 vehículos autónomos fue lanzada por Google al tránsito diario para ser evaluados fuera de las pistas de pruebas. En la actualidad, se pueden ver coches de Google circulando de manera autónoma. Sin embargo, en la actualidad el vehículo más conocido como coche autónomo es probablemente el Tesla Model S, Figura 3, el cual está dotado del sistema Autopilot 2.0, que consta de 8 cámaras, 12 sensores ultrasónicos con un alcance de 8 metros y de un radar frontal con un alcance de 160 metros. Todo esto controlado por un sistema NVIDIA lo convierten en un sistema de inteligencia artificial de aprendizaje neuronal, el cual tiene capacidad para aprender a reconocer patrones de forma rápida. Sin embargo, pese al gran trabajo conseguido por el fabricante Tesla Motors, en mayo de 2016 se produjo un accidente de un

Modelo S con la función de Autopilot activada, produciéndose la primera muerte con el piloto automático activado en un vehículo. Es por esto que todavía queda un largo camino a recorrer mejorando la conducción autónoma para conseguir que sea una realidad de nuestro día a día.

1.2. OBJETIVOS

El objetivo principal de este trabajo es el de implementar, en la nueva arquitectura de software del vehículo inteligente IVVI 2.0 [5], el algoritmo de asistencia a la conducción basado en la detección y clasificación de la calzada.

1.2.1. IVVI 2.0

El IVVI 2.0 (Figura 4) es la segunda plataforma de investigación del Laboratorio de Sistemas Inteligentes para sistemas basados en visión por computador, acompañados por técnicas láser. El objetivo de este vehículo es la



Figura 4: Vista exterior del IVVI 2.0 desarrollado por la UC3M

investigación y testeo de nuevos algoritmos en condiciones reales. Actualmente, se están probando sistemas como la detección de peatones mediante el uso de

cámaras térmicas, detección de señales de tráfico, detección de otros vehículos, monitorización del conductor para detectar somnolencia o fatiga, etc.

Los diferentes dispositivos integrados en el IVVI 2.0 incluyen un sistema de visión estéreo, Figura 5, para la detección y clasificación de las líneas de la carretera, y la detección de objetos tales como vehículos, peatones, etc., en entornos complejos. Una de las cámaras estéreo es utilizada para la detección de las señales de tráfico verticales. Incluye también una cámara infrarroja para la detección de peatones por la noche o en condiciones de poca visibilidad. Un sistema Kinect, instalado en el salpicadero del IVVI 2.0, reconoce la cara del conductor y monitoriza sus movimientos para controlar los niveles de somnolencia. En el paragolpes delantero se encuentra un láser multicapa para la detección de objetos, y por último un receptor GNSS (Sistema Global de Navegación por Satélite) y un medidor de inercia, los cuales se encuentran en una plataforma en el techo del vehículo, alineada con el eje trasero.



Figura 5: Sistema estéreo y cámara infrarroja utilizados para monitorizar la carretera

Un ordenador se encarga de procesar toda la información recopilada por estos sensores y cámaras, en tiempo real. Con el fin de mantener al conductor

informado, este sistema se encuentra conectado con los altavoces del vehículo, a través del cual se reproducen distintos tipos de alertas y mensajes. Además, en una pantalla situada en el salpicadero muestra alertas de peligro.

Recientemente se ha cambiado la arquitectura de software del ordenador de este vehículo, pasando a utilizar la arquitectura *Robot Operating System* (ROS) ya que este proporciona múltiples ventajas respecto a la anterior arquitectura del IVVI 2.0, destacando sobre todo la posibilidad de reutilizar programas, ya que ROS permite mucha más comunicación entre distintos módulos, con lo que se reutiliza código y por tanto se optimiza el rendimiento del ordenador y se consiguen mejores resultados.

1.3. ESTRUCTURA DEL CONTENIDO

El resto del contenido se organiza de la siguiente manera: en el Capítulo 2 se expone el estado del arte, es decir, la situación actual en cuanto a sistemas ADAS, y más concretamente, la situación actual del Sistema de Detección de Carril, en el Capítulo 3 se explicarán los recursos utilizados durante el desarrollo del trabajo, en el Capítulo 4 se encuentra la descripción general del sistema, en el Capítulo 5 se expone los nodos que han sido implementados o modificados para conseguir el objetivo principal y el correcto funcionamiento del sistema, en el Capítulo 6 se muestran los resultados obtenidos al utilizar este Sistema de Detección de Marcas Viales en diferentes situaciones, en el Capítulo 7 se expone el marco regulador actual para la circulación de estos vehículos tanto en nuestro país como en Estados Unidos, ya que es el país más importante en el uso de vehículos autónomos, en el Capítulo 8 se expondrá el Entorno socio-económico en el que se desarrollará el impacto socio-económico de los sistemas ADAS y más concretamente, del sistema de detección de marcas viales, y también se realizará un presupuesto con la estimación del coste de este trabajo, y por último en el Capítulo 9 se expondrán las conclusiones sacadas en la realización de este trabajo.

También se expondrán ideas a implementar en trabajos futuros con el fin de mejorar este sistema.

CAPÍTULO 2: ESTADO DEL ARTE

En este capítulo se describe el estado actual del sistema que se va a implementar en este proyecto. Para ello es necesario citar los principales sistemas de ayuda a la conducción (ADAS).

2.1. SISTEMAS DE AYUDA A LA CONDUCCIÓN

En la actualidad existen numerosos sistemas ADAS. Algunos muy conocidos, ya que se encuentran implementados en la mayoría de los vehículos, tales como el sistema de detección de ángulo muerto o el sistema de ayuda al aparcamiento. Sin embargo, con el avance de la tecnología se han implementado otros sistemas más sofisticados que están ganando mucha popularidad. Estos sistemas están basados en visión por computador, lo que refleja la importancia que ha adquirido la utilización de cámaras y sensores en la automoción. A continuación, se muestran algunos ejemplos de estos sistemas ya comercializados:

2.1.1. Sistema de visión nocturna de Mercedes:

Gracias a una cámara infrarroja, junto con la ayuda de dos faros, también de este tipo, los cuales emiten un haz de luz, invisibles para el ojo humano, es capaz de detectar objetos calientes, como personas o animales, así como objetos fríos, clara ventaja frente a los sistemas térmicos convencionales, los cuales no son capaces de detectar objetos fríos, Figura 6. Consultar [6] para más información.



Figura 6: Visualización del Sistema de Visión Nocturna de Mercedes en el salpicadero.

Otros sistemas como el de Audi [7], cuando detecta un objeto caliente, lo muestra en la pantalla del salpicadero con un recuadro amarillo, y si considera que están en una zona con riesgo de atropello, los marca en rojo y suena un aviso.

2.1.2. Control de cruce adaptativo (ACC) de Bosch:

Este sistema regula la velocidad del vehículo manteniendo la distancia de seguridad con el vehículo precedente. Un sensor de radar de larga distancia controla el tráfico delante del vehículo, Figura 7, ya que, a través de las señales recibidas, calcula la distancia, la dirección y la velocidad relativa de los vehículos precedentes, aunque la carretera no siga una línea recta.

El sistema puede actuar tanto directamente en el control del motor, como en el sistema de frenos en combinación con el programa electrónico de estabilidad (ESP). Además, este sistema se puede mejorar mediante la adición de sensores

adicionales, como una cámara de video u otros sensores de radar de corta o larga distancia [8].

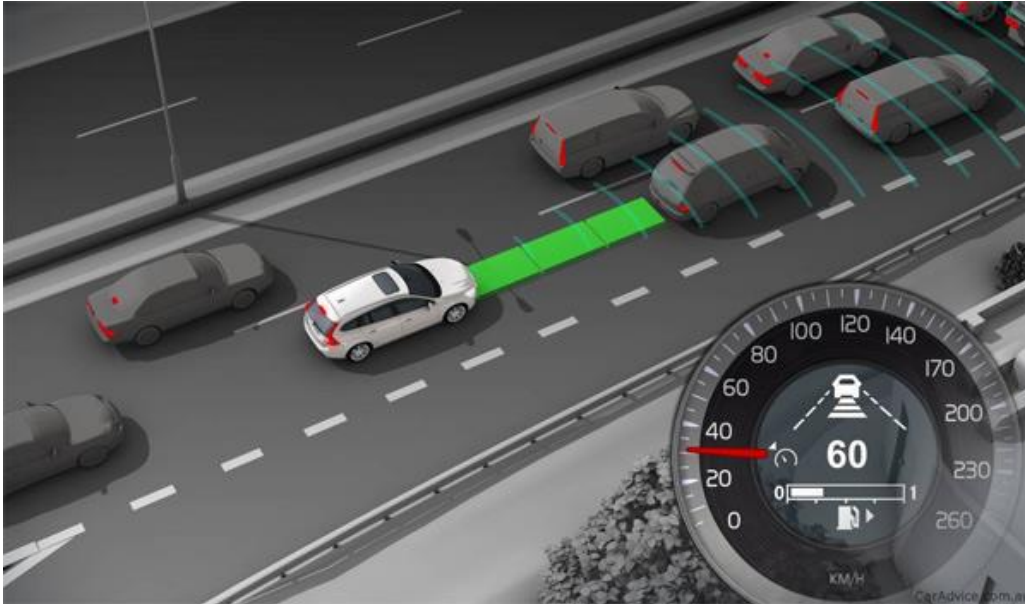


Figura 7: Representación del Control de Crucero Adaptativo

2.1.3. Detector de fatiga del conductor:

Con el fin de reducir el número de accidentes debidos a la fatiga, Volkswagen incorpora el Detector de Fatiga de Bosch [9], por tanto, se trata de un sistema preventivo. Dicho sistema incorpora una cámara de reconocimiento facial la cual enfoca a la cara del conductor y controla los ojos de este. De esta manera es capaz de saber si el parpadeo indica sueño, si el conductor mira a la carretera o por el contrario el conductor está distraído.

Este sistema es capaz de reconocer otros gestos como bostezos u otras expresiones que muestren cansancio, pero también es capaz de reconocer si el conductor está estresado y nervioso, Figura 8. El sistema avisa al conductor a través de una alarma, luces o mensajes de texto en el cuadro de instrumentos, instándole a no distraerse, parar para descansar o calmarse.

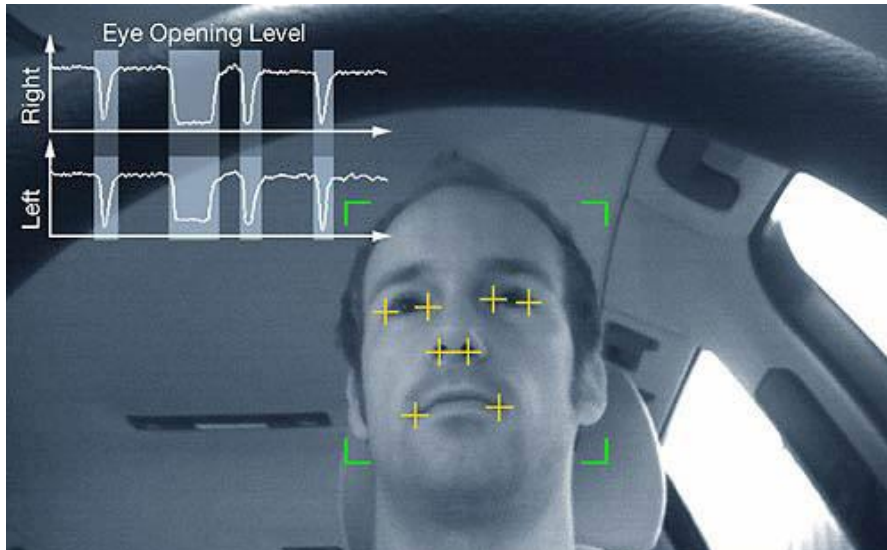


Figura 8: Ejemplo de un Sistema de Detección de fatiga.

2.1.4. Sistema de reconocimiento de señales:

Se trata de un sistema basado en la visión por computador, el cual es capaz de identificar y clasificar señales de distintos tipos, tales como de velocidad, prohibiciones de adelantamiento o tramos de obras, Figura 9. Estas señales se muestran en la pantalla del salpicadero, y algunos sistemas incluso emiten alarmas cuando el conductor excede la velocidad máxima detectada por el vehículo.



Figura 9: Sistema de Detección de Señales Verticales de Volvo

2.1.5. Detector de peatones con frenada de emergencia:

En 2015 fallecieron 367 peatones en accidentes de tráfico, Tabla 1, según la DGT [2], lo que supone el 22% del total de fallecidos en accidentes de tráfico, porcentaje superior en dos puntos al del año 2014, Tabla 11.

Tipo de vía	Fallecidos		Heridos hospitalizados		Heridos no hospitalizados		Índice de letalidad
	Número	%	Número	%	Número	%	
Vías interurbanas	120	33%	236	12%	483	4%	14,3
Vías urbanas	247	67%	1.763	88%	11.673	96%	1,8
Total	367	100%	1.999	100%	12.156	100%	2,5

Tabla 1: Tabla de peatones fallecidos, heridos hospitalizados, heridos no hospitalizados e índice de letalidad. Vías interurbanas y urbanas. España, 2015

Años	Vías interurbanas		Vías urbanas		Total	
	Peatones fallecidos	% Peatones fallecidos sobre el total	Peatones fallecidos	% Peatones fallecidos sobre el total	Peatones fallecidos	% Peatones fallecidos sobre el total
2006	317	9%	296	40%	613	15%
2007	287	9%	304	41%	591	15%
2008	236	10%	266	42%	502	16%
2009	201	9%	269	46%	470	17%
2010	193	10%	278	51%	471	19%
2011	158	10%	222	49%	380	18%
2012	144	10%	232	50%	376	20%
2013	154	13%	224	50%	378	23%
2014	132	11%	204	46%	336	20%
2015	120	10%	247	56%	367	22%

Tabla 2: Tabla de evolución de los peatones fallecidos. Vías interurbanas y urbanas. España, 2006-2015

Este sistema busca evitar dichos accidentes o si es posible minimizar los daños reduciendo la velocidad a la que se produce la colisión.

Normalmente estos sistemas están compuestos por una cámara, para detectar al peatón, y un radar, para calcular la distancia del peatón detectado respecto al vehículo, Figura 10. Una vez detectado el peatón, si el conductor no acciona el freno, el coche lo activará de manera autónoma para evitar la colisión.

Actualmente hay sistemas más modernos los cuales son capaces no solo de detectar peatones que se encuentran delante del vehículo, sino que son capaces incluso de detectar a personas que se disponen a cruzar en la trayectoria del coche.

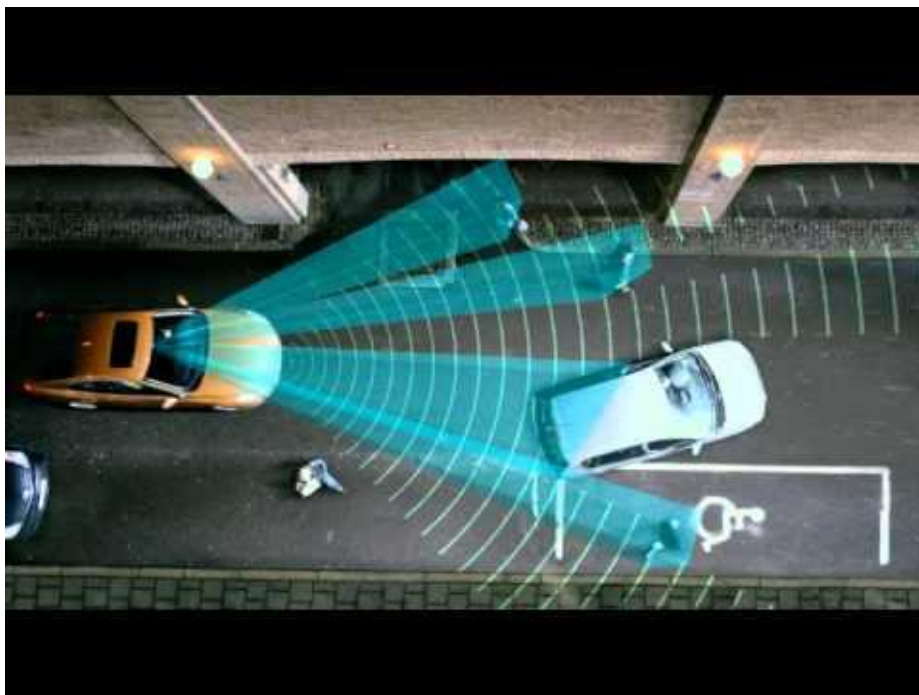


Figura 10: Sistema de detección de peatones de Volvo, 2017

Tanta es la popularidad que está ganando este sistema que Euro NCAP, programa de seguridad para automóviles, ha añadido a sus pruebas estos sistemas de detección de peatones con frenada autónoma de emergencia (AEB).

Además de estos productos ya comercializados existen innumerables trabajos académicos donde se investigan y desarrollan otros sistemas ADAS. Algunos ejemplos son:

* Tesis Doctoral – Agentes de control de vehículos autónomos en entornos urbanos y autovías [5]:

En dicha tesis se busca mejorar la seguridad y el confort en el control de vehículos autónomos, pudiendo realizar maniobras tales como adelantamientos, marcha atrás, incorporaciones o maniobras dentro de rotondas, en diferentes escenarios urbanos, así como en autovías.

Para conseguir esto se basa en el control lateral y longitudinal utilizando técnicas de control básico, así como técnicas de inteligencia artificial, fundamentalmente control borroso y neuro-borroso.

* Proyecto de Grado – Diseño, construcción e implementación de un sistema de reconocimiento de objetos en carretera durante la conducción, bajo condiciones adversas, en un vehículo [6]:

El objetivo principal de este sistema ADAS es el de proporcionar mayor seguridad cuando se conduce un vehículo en condiciones adversas tales como lluvia, niebla u otras condiciones de baja visibilidad.

El sistema propuesto utiliza un módulo de control electrónico que procesa la información recibida a través de un sensor laser de largo alcance ubicado en la parte frontal del vehículo, pudiendo detectar objetos que se encuentran en la carretera y notificar la presencia de estos al conductor, incluso actuar en la mecánica del coche para llegar a detenerlo en caso de colisión.

2.2. AVISO O ASISTENCIA DE MANTENIMIENTO DE CARRIL:

Este proyecto se centrará en el desarrollo de este tipo de asistencia a la conducción. El sistema de reconocimiento de carril se trata de un sistema electrónico, incluido en las características comúnmente conocidas como ADAS

[8], que reconoce las líneas del carril por el que circula el coche, y en caso de que este se desvíe de la trayectoria y pise o sobrepase alguna de las líneas de manera involuntaria, es decir, por la ausencia de señalización de un cambio de carril con el intermitente correspondiente, dependiendo del sistema, actúa sobre la dirección de este para corregir la trayectoria, o simplemente hace saltar una alerta al conductor, normalmente haciendo vibrar el volante.

Algunas marcas usan sensores infrarrojos para detectar las líneas, los cuales se encuentran normalmente en el paragolpes delantero del vehículo y los cuales, al estar enfocados hacia la calzada, reconocen las líneas del carril, Figura 11. Sin embargo, este sistema ha ido evolucionando hasta el uso de una cámara situada detrás del parabrisas, junto al espejo retrovisor, la cual es capaz mediante software de reconocer el carril. Ver [13]

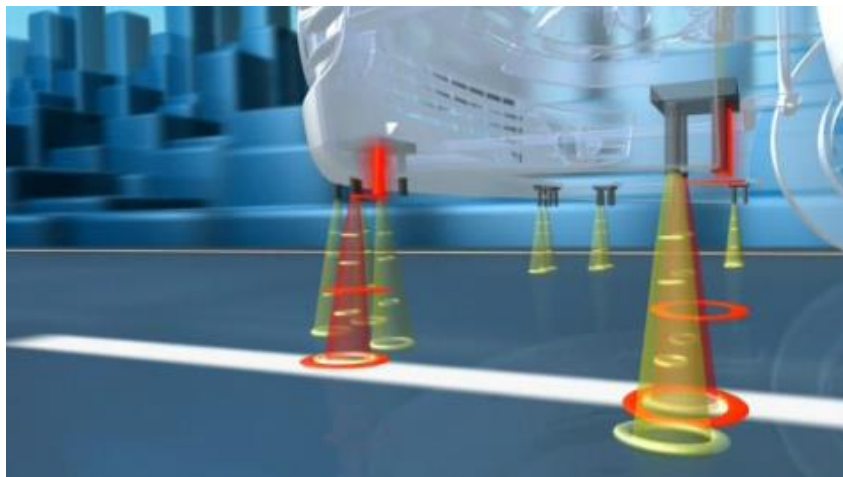


Figura 11: Ejemplo de Sistema de aviso de salida involuntaria de carril mediante la colocación de sensores en el parachoques delantero

El primer indicio de un sistema de detección de carril es en el año 2001 cuando Nissan Motors sacó al mercado japonés el Cima, el cual contaba con un sistema de mantenimiento de carril. En 2002, Toyota también introduce un sistema parecido en modelos como el Caldina y Alphard, ambos vendidos en Japón. En 2003 aparece el primer vehículo, con un sistema de este tipo, basado en la detección del carril mediante el uso de una cámara estéreo. Se trataba del

modelo Inspire de Honda el cual montaba una cámara en la parte superior del parabrisas.

En 2004, Iteris y Valeo desarrollan conjuntamente el primer sistema de vehículos de pasajeros disponible en Norteamérica. En este sistema, una cámara busca las marcas del carril y se activa un tono de advertencia cuando el vehículo se desplaza sobre las líneas detectadas.

Citroën, en 2005, se convirtió en el primer fabricante en ofrecer un sistema de detección de carril en Europa. Este sistema utilizaba sensores de infrarrojos para monitorear las marcas del carril, y un sistema de vibración en el asiento del conductor alertaba a este de las desviaciones.

Durante los siguientes años y hasta la actualidad marcas como Lexus, Mercedes-Benz, Audi, Figura 12, BMW, Volvo y General Motors han ido introduciendo sistemas de aviso de salida de carril, la mayoría de ellos basados en la tecnología de núcleo de Mobileye, empresa dedicada a la visión por computador.

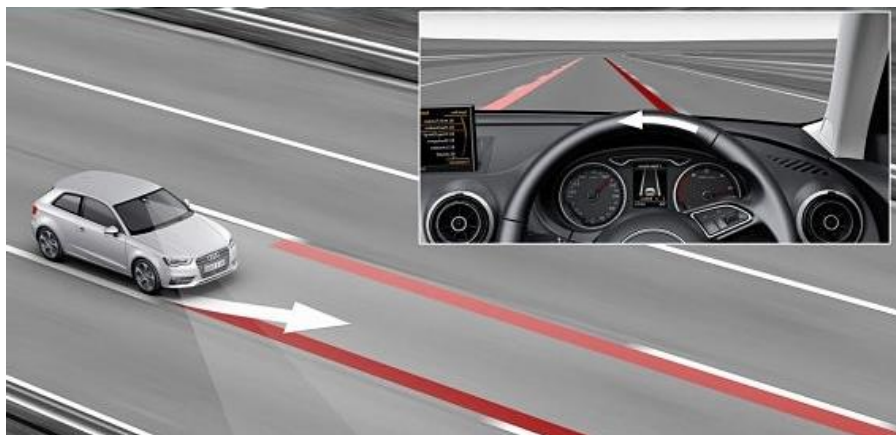


Figura 12: Sistema de detección de carril con control sobre la dirección de Audi

Si bien estos sistemas eran utilizados como una ayuda a la conducción y no como conducción autónoma, ya que únicamente corregían la trayectoria para

evitar una salida involuntaria de carril, ya existen algunos modelos que, con técnicas mejoradas de detección de carril, son capaces de conducir de manera autónoma, como los últimos modelos de Tesla.

En cuanto a antecedentes académicos, existen algunos trabajos para la implementación de sistemas de detección de carril. En la tesis doctoral “Detección y Modelado de Carriles de Vías Interurbanas mediante Análisis de Imágenes para un Sistema de Ayuda a la Conducción” [7], con resultados positivos.

Los tres métodos más utilizados para llevar a cabo la detección de la carretera son:

- Basados en el aprendizaje: Son métodos muy eficientes en tareas concretas, mientras que los principales inconvenientes son que requieren de un conjunto representativo de patrones de entrenamiento, difícil de conseguir, y que, debido a la variabilidad de la carretera, tienen una capacidad de generalización muy limitada.
- Basados en la extracción de características de la imagen: Tienen la ventaja de detectar todos los elementos de la imagen, pudiendo identificar todas las líneas. Sin embargo, esta ventaja es al mismo tiempo una desventaja, ya que pueden producirse con facilidad falsos positivos, y requieren por tanto un filtrado de líneas complejo.
- Basados en el ajuste de un modelo: Parten de un modelo predefinido que se ajusta a las observaciones. Su inconveniente principal es que únicamente se identificarán aquellos objetos que se contemplen en el modelo.

Un ejemplo de alguno de estos sistemas es el proyecto ALVINN [15], el cual utilizaba métodos basados en aprendizaje, Figura 13.



Figura 13: Vehículo utilizado durante el proyecto ALVINN.

CAPÍTULO 3: RECURSOS UTILIZADOS

3.1. ROS

3.1.1. INTRODUCCIÓN:

ROS es un espacio de trabajo para el desarrollo de software para robots. Como meta-sistema operativo que es, ya que funciona por encima de otro sistema operativo (GNU/Linux), provee servicios estándar como abstracción del hardware, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes, [10]. También proporciona herramientas y librerías para obtener, compilar, escribir y ejecutar código en distintos ordenadores. ROS está compuesto por nodos independientes.

Los nodos de ROS no tienen que estar en el mismo sistema, incluso pueden tener distintas arquitecturas, lo que convierte a ROS en una infraestructura muy flexible y adaptable a diversas necesidades.

ROS se desarrolló originalmente en 2007 por el Laboratorio de Inteligencia Artificial de Stanford (SAIL). En 2008 se continuó el desarrollo en Willow Garage, instituto de investigación robótico con más de veinte instituciones colaborando en el desarrollo. En 2013, ROS se transfirió a la *Open Source Robotics Foundation*.

En la actualidad ROS solo es posible ejecutarlo en sistemas operativos basados en UNIX, siendo probado esencialmente en Ubuntu y Mac OS X. Actualmente se está intentando incluir ROS en la plataforma WINDOWS.

3.1.2. CONCEPTOS:

ROS es una red peer-to-peer (red de pares), es decir, es una red de ordenadores que funciona por una serie de nodos que se comportan como iguales entre sí. No utilizan ni

clientes ni servidores fijos. En la Figura 14 se puede ver un esquema de este modo de funcionamiento. Por tanto, los datos de ejecución vienen dados por los siguientes conceptos:

- *Nodos*: Los nodos son ejecutables que se comunican con otros procesos a través de tópicos, servicios y servidores de parámetros. ROS está diseñado para ser modular en una escala de grano fino, un sistema de control de robots comprende usualmente muchos nodos. Así en este trabajo se utilizarán varios nodos como por ejemplo uno encargado de realizar la disparidad, otro para la calibración de la imagen, etc. Más adelante se hablará de estos nodos en profundidad.

Un nodo tiene que tener un nombre único en el sistema y puede ser escrito usando diferentes librerías como *roscpp* (para C++) y *rospy* (para Python).

ROS tiene herramientas para manejar los nodos. La principal herramienta es *roscpp*, la cual es una herramienta de línea de comando para mostrar información sobre los nodos:

Una forma de ejecutar varios nodos al mismo tiempo e incluso pasar a estos parámetros, o realizar cambios de nombre, etc., es a través de un archivo conocido como ***launch***, el cual es una herramienta muy útil cuando se van a realizar varios procesos a una imagen, como es el caso de este trabajo.

- *Master*: El ROS Master proporciona el registro de nombres y la búsqueda para el resto de la ejecución de ROS. Sin el Master, los nodos no serían capaces de encontrar al resto de nodos, intercambiar mensajes o invocar servicios.

- *Servidor de parámetros*: posibilita guardar datos en una localización central. En la actualidad es parte del Master.

- *Mensajes*: Los nodos se comunican entre sí a través de los mensajes. Un mensaje es simplemente una estructura de datos compuestos. Un mensaje es una combinación de tipos primitivos y mensajes. Los tipos primitivos de datos

soportados son entero, flotante, booleano, etc., así como los *arrays* de tipo primitivo. Los mensajes pueden incluir estructuras y *arrays* (como las estructuras de C).

- *Tópicos*: Los mensajes son enrutados por un sistema de transporte que utiliza una semántica publicador/subscriptor. El tópico tiene un nombre para identificar el tipo de mensaje. Cualquier nodo puede subscribirse a el tópico y a través de dicho nombre publicar mensajes de distintos tipos, o bien extraer información del tópico para utilizarla en el nodo. Pueden existir distintos publicadores para el mismo tópico, al igual que un nodo puede publicar o subscribirse a diversos tópicos, la única restricción es que el nodo y el tópico tienen que tener el mismo tipo de mensaje.

ROS tiene una herramienta para trabajar con tópicos llamada *rostopic*. Se trata de una herramienta de línea de comandos muy útil la cual muestra información sobre el tópico o publica datos directamente sobre la red.

- *Servicios*: Aunque los tópicos ofrecen multitud de posibilidades de comunicación entre nodos, cuando necesitas comunicarte con otros nodos y recibir una respuesta, se utilizan los servicios, los cuales se definen a partir de una estructura de mensajes, una para la petición y otra para la respuesta. Un nodo proporcional ofrece un servicio bajo un nombre y un cliente utiliza el servicio enviando el mensaje de solicitud y esperando la respuesta.

Existe además la posibilidad de introducir parámetros en el servidor de tal manera que los nodos puedan acceder a estos y recibir parámetros en tiempo de ejecución. ROS tiene una serie de herramientas para trabajar con el servidor de parámetros, como por ejemplo “*rosethparam list*”, “*rosethparam get*”, etc.

- *Bolsas*: Para guardar y reproducir de nuevo mensajes de ROS se utilizan las bolsas. Estas son un mecanismo importante para el almacenamiento de datos, como lecturas de sensores, que pueden ser difícilmente adquiridas pero que son

necesarias para el desarrollo y testeo de algoritmos. Son reconocidos mediante el formato *.bag*.



Figura 14: Esquema conexión entre nodos, tópicos y servicios

3.2. OPENCV

OpenCV [17] es una librería de computación visual creada por Intel. Desde que apareció ha sido utilizada en múltiples aplicaciones, desde sistemas de seguridad con detección de movimiento, hasta aplicaciones que requieren de detección de objetos. OpenCV está disponible para multitud de plataformas como Windows, Linux, Mac, Android, etc. Además, soporta diferentes lenguajes de programación, como Python, Java, C++ entre otros.

Uno de los objetivos de OpenCV es crear una infraestructura de visión por computador fácil de utilizar y al mismo tiempo que permita a los programadores desarrollar aplicaciones complejas.

OpenCV está estructurado de manera modular. Los módulos principales de OpenCV son:

- *core*: Es el módulo básico y por tanto es en el que se encuentran las funciones básicas de procesamiento de imágenes.

- *highhui*: Este es el módulo que provee la interfaz de usuario, así como la capacidad de capturar imágenes y video.

- *imgproc*: En este módulo es en el que se encuentran los algoritmos de procesamiento de imágenes, entre los que se incluyen filtrado de imágenes, transformado de imágenes, etc.

- *video*: En este módulo de análisis de video se encuentran otros algoritmos como seguimiento de objetos, entre otros.

- *objdetect*: Incluye algoritmos de detección y reconocimiento de objetos estandar.

La clase `cv::Mat` es la parte fundamental de la biblioteca OpenCV, la cual nos posibilita guardar y manipular los píxeles de una imagen. La forma de almacenar dichas imágenes es a través de matrices, donde en cada celda de la matriz se almacenará un valor numérico el cual representará el color de dicho píxel en la imagen.

3.3. POINT CLOUD LIBRARY (PCL)

Se trata de una biblioteca de código abierto que se utiliza para tareas de procesamiento de nubes de puntos. Esta biblioteca contiene algoritmos capaces de realizar tareas como reconstrucción superficial, registro, ajuste de modelos o segmentación. PCL [8] surgió ante la necesidad de que los robots pudieran percibir el mundo de la misma manera que lo hacemos los seres humanos.

Esta biblioteca se encuentra dividida en una serie de módulos para facilitar su compilación. De los distintos módulos caben destacar los siguientes:

- *Filtros*: En este módulo se encuentran una serie de herramientas que permiten eliminar datos atípicos con el fin de obtener mayor fiabilidad en los

procesos a desarrollar. También existe la posibilidad de utilizar algunos filtros para reducir el volumen de datos que genera una nube de puntos, permitiendo así una mayor rapidez computacional.

- *Octree*: Proporciona métodos eficaces para dividir nubes de puntos en conjuntos para realizar procesos de forma más eficiente. Esta biblioteca ofrece rutinas eficientes de búsqueda de vecinos. Estos algoritmos pueden ajustarse automáticamente a la dimensión del conjunto de puntos.
- *Características*: Permite estimar parámetros para determinar si un conjunto de datos representa una función geométrica en el espacio. Esta biblioteca también permite extraer características tridimensionales.
- *Segmentación*: Como su propio nombre indica, esta biblioteca contiene algoritmos desarrollados para la segmentación de nubes de puntos.

3.4. GIT

Git es un software de control de versiones (VCS), de modo que Git permite recuperar versiones anteriores de cualquier archivo, como cualquier otro controlador de versiones. Sin embargo, tal y como expone [9] la principal diferencia entre Git y cualquier otro controlador es cómo Git modela sus datos. Git modela sus datos como un conjunto de instantáneas, es decir, cuando se confirma un cambio, Git realiza una captura del aspecto de todos los archivos en ese instante, y guarda una referencia a esa captura. Esta es la principal distinción entre Git y prácticamente todos los demás VCSs.

Git tiene tres estados en los que pueden estar los archivos:

- *Confirmado (committed)*: Los datos que se encuentren en este estado están almacenados de manera segura en la base de datos.

- *Modificado (modified)*: Este estado significa que el archivo ha sido modificado, pero todavía no ha sido confirmado a la base de datos.

- *Preparado (staged)*: Un archivo modificado ha sido marcado en su versión actual para ser añadido a la base de datos en la próxima confirmación.

Estos tres estados están relacionados con las tres secciones principales de las que consta cualquier proyecto de Git:

- *Directorio de Git (repository)*: Es donde Git almacena los metadatos y la base de datos del proyecto. Es, por tanto, la parte más importante de Git. Esta base de datos es la que se copia cuando se realiza una clonación del repositorio.

- *Directorio de trabajo*: Es una copia de una versión del proyecto. Estos archivos son copiados de la base de datos.

- *Área de preparación (staging area)*: Es un archivo sencillo que almacena información acerca de lo que va a ir en la próxima confirmación.

La Figura 15 muestra un esquema general de este funcionamiento.

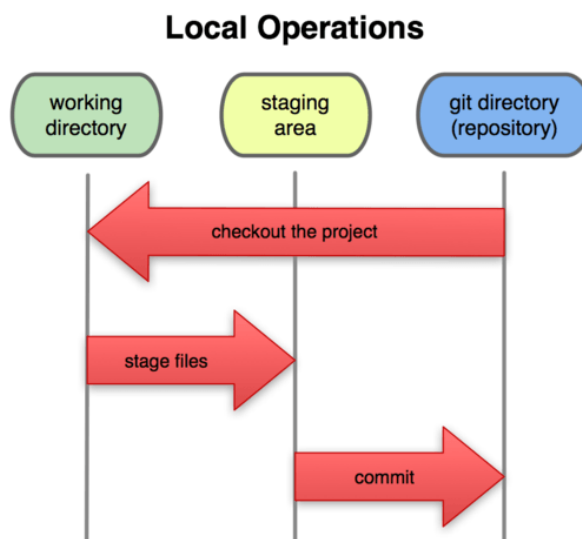


Figura 15: Esquema general del funcionamiento de secciones principales de Git y sus estados

CAPÍTULO 4: DESCRIPCIÓN GENERAL DEL SISTEMA

4.1. IMPLEMENTACIÓN DEL ALGORITMO DE DETECCIÓN DE CARRIL

A través del archivo “*tfg_launch*”, de tipo “*launch*”, se ejecutan todos los nodos con los diferentes procesos que se van a aplicar a cada imagen para conseguir detectar el carril. A continuación, se explican los diferentes nodos o programas necesarios para completar el objetivo de este trabajo. En la Figura 16 se observa el esquema de funcionamiento del sistema.



Figura 16: Esquema general nodos

4.1.1. DISPARIDAD:

La Real Academia Española (RAE) define la disparidad como “Desemejanza, desigualdad y diferencia de unas cosas respecto de otras”. Como su propia definición dice, al aplicar esto a una imagen, lo que buscamos es encontrar el conjunto de diferencias entre píxeles para poder clasificarlos y crear un entorno 3D a partir de un par estéreo.

La visión humana gracias a la posición de los ojos y la forma de moverlos es capaz de obtener una imagen en 3D, aunque aparentemente las imágenes que reciben en cada ojo son prácticamente iguales, con la diferencia en la posición relativa de los ojos. Estas diferencias son lo que conocemos como disparidad, y existe una relación directa con la distancia o profundidad a la que se encuentran los objetos respecto al observador. Se llama visión estéreo a la capacidad de recuperar la estructura tridimensional de una escena a partir de dos imágenes diferentes de la misma. Una formulación más cercana al problema computacional es la de localizar para cada punto de una imagen su homólogo en la otra imagen. De esta forma se puede conocer la profundidad relativa de dichos puntos.

Por tanto, una forma para estimar la profundidad de los puntos en una determinada escena es calculando la disparidad entre las distintas imágenes de la misma escena. Para esto supondremos que los objetos captados en la escena no modifican su posición ni sufren deformaciones.

En este proyecto se han utilizado dos cámaras de características teóricamente idénticas, y se han configurado como se muestra en la Figura 17(a). Estas dos cámaras forman un par estéreo. Asumiendo que cada una de las cámaras cumplen el modelo *pinhole*, ya que ambas han sido calibradas para que así sea, se cumplen una serie de requisitos: Los ejes ópticos de las cámaras son paralelos, $\overrightarrow{O_R o_R} \parallel \overrightarrow{O_L o_L}$; Ambas tienen la misma distancia focal (f); Las imágenes que forman cada una de las cámaras, I_L e I_R , están en planos paralelos, con centros

O_L y O_R separados por una distancia B , conocida como línea base B . De este modo el eje \vec{x} y la línea base son paralelas. Habiendo considerado el modelo pinhole, tomando un punto cualquiera del espacio, P , con coordenadas $(X, Y, Z, 1)^T$, es proyectado en cada una de las imágenes en los puntos p_L y p_R , con coordenadas homogéneas $(x_L, y_L, 1)^T$ y $(x_R, y_R, 1)^T$, respectivamente. En la Figura 17(b) se ve la relación de los parámetros definidos en el par estéreo, los cuales permiten obtener la relación entre la disparidad d y la profundidad Z del punto P .

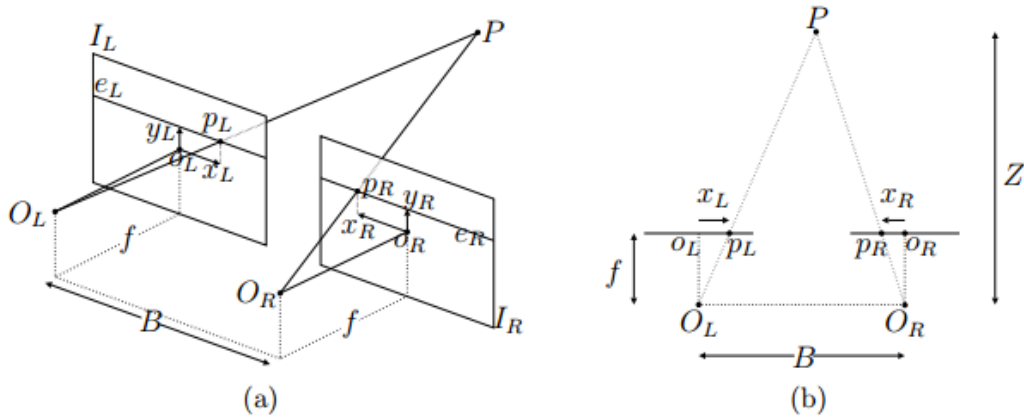


Figura 17: Configuración de las cámaras del par estéreo. (b) Relación entre los parámetros para obtener la profundidad Z a partir de la disparidad d .

Existen distintas técnicas para calcular el mapa de disparidad. En el archivo `.launch "disparity_color"`, situado en el paquete `"disparity"`, encargado de calcular la disparidad, existe un parámetro denominado `"method"` el cual selecciona el tipo de disparidad a utilizar entre las dos opciones ofertadas. Estas son *block matching* y *semi-global matching*. Por defecto se utilizará *block matching* [13].

La técnica de *block matching* consiste en la búsqueda de un objeto conocido en una imagen. La forma de utilizar esta técnica es la siguiente:

- Inicialmente se recorre una de las imágenes del par estéreo, creando una plantilla.

- A continuación, se establece un rango de búsqueda de correspondencia. De esta manera en un sistema con cámaras paralelas como el propuesto, se tiene además la información de que un píxel de la imagen izquierda tiene su correspondencia en la imagen derecha, siempre en una posición más a la izquierda.

- Dentro de este rango de búsqueda, se analiza la vecindad de cada píxel con la plantilla creada.

- Por último, el píxel que tenga mayor similitud será el considerado como la correspondencia con el píxel en análisis.

El resultado de disparidad, es decir la profundidad de cada uno de los píxeles de la escena es publicada en el tópico “*stereo_camera/disparity*”.

4.1.2. NUBE DE PUNTOS:

La definición más sencilla de una nube de puntos es la de un conjunto de puntos 3D de igual tamaño a la cantidad de píxeles de cada imagen, obtenida a partir de un par estéreo, o a partir de una Kinect, láser, etc. De esta manera, por ejemplo, para una resolución típica de 480 líneas horizontales por 640 líneas verticales, el número de puntos en la nube de puntos será de 307 200. El número de puntos que forman la nube de puntos depende de la resolución y la precisión de la cámara, es decir, del número de píxeles de la imagen. Ver ejemplo en la Figura 18.



Figura 18: Comparación de la PC con la imagen original

Las nubes de puntos requieren en muchos casos de una gran cantidad de memoria ya que son estructuras de gran tamaño. Además, su manera de organización y su gran volumen de datos genera altos tiempos y costes de computación. Dado que la ejecución de los algoritmos para la asistencia a la conducción han de ser ejecutados en tiempo real, es necesario reducir la nube de puntos, sin afectar a su uniformidad espacial, reduciendo así el tiempo de búsqueda de procesos posteriores, como la extracción del plano.

Para realizar esta simplificación del entorno 3D se ha utilizado un método conocido como método *Voxel* [21], el cual consiste en dividir la nube de puntos en una rejilla tridimensional de cubos con un tamaño fijo. A continuación, dentro de cada cubo se calcula el centroide de todos los puntos contenidos en dicho cubo. Este punto es la simplificación de todos los puntos del cubo. Al realizar esta acción en cada cubo, el resultado es una nube de puntos simplificada. Como se ve en el ejemplo de la Figura 19,

la uniformidad del PC se verá afectada en función del tamaño de los cubos utilizados para segmentar el entorno.

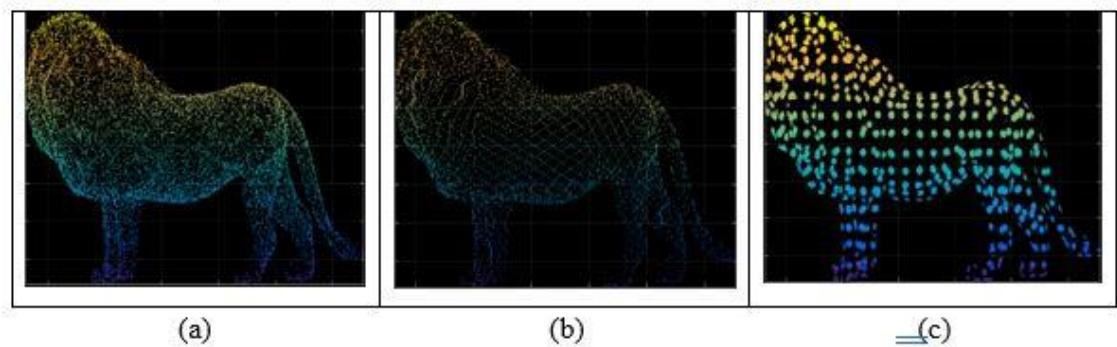


Figura 19: Ejemplo evolución uniformidad de la PC. (a) PC sin método Voxel; (b) PC con tamaño de cubos Voxel medio; (c) PC con tamaño de cubos Voxel alto

El principal objetivo de realizar esta nube de puntos es el de conseguir una característica significativa para poder diferenciar el plano de la carretera respecto al resto de la escena, ya que siempre la carretera es el plano con una cantidad mayoritaria de puntos o PC.

La nube de puntos calculada es publicada, por el nodo o paquete “*point_cloud*”, en el tópico “*/stereo_camera/points2*”, y a través de un visualizador como *Rviz* es posible verla subscribiéndose a ese tópico.

4.1.3. EXTRACCIÓN DEL PLANO Y OBTENCIÓN DE PARÁMETROS

EXTRÍNSECOS:

Como el propio nombre indica, la función principal de este nodo es la de identificar el plano de la carretera y separarlo del resto para conseguir la vista de pájaro (*bird view*), la cual se explicará más adelante.

Existen distintos métodos para la obtención o estimación de dicho plano; a estos métodos se les llama estimadores. Los estimadores más utilizados son *LMedS* [14] y el estimador RANSAC [23], el cual será el utilizado en este proyecto.

RANSAC es un estimador robusto desarrollado en 1981 basado en la aplicación de una técnica de votación a partir de un muestreo aleatorio, con el objetivo de determinar el número de observaciones erróneas “*outliers*” y el número de observaciones válidas “*inliers*”.

Este método consta de dos pasos los cuales se repiten constantemente hasta llegar alcanzar un resultado óptimo. Estos pasos son:

1. Generación de la hipótesis: En este paso se selecciona un conjunto de muestras mínimas aleatorio, al que llamaremos MSS (*Minimal Sample Set*), del conjunto de datos totales.

2. Prueba: En este paso se escoge un subconjunto de datos los cuales están dentro de una distancia umbral T del modelo. Si el número de datos contenidos en el subconjunto es mayor que el número de datos contenido en MSS es remplazado por este y se repite el procedimiento.

Para determinar el número de veces que se debe escoger un MSS, se calcula a partir de la probabilidad de que al realizar m experimentos al menos un MSS genere un modelo correcto, [15].

El MSS resultante es el que define el plano de la carretera. A través de una serie de procesos matemáticos explicados en [15] se extraen los parámetros, Figura 20, que definen un plano, así como el cabeceo (α) y el alabeo (σ), *pitch* y *roll* [24].

$$ax + by + cz + d = 0 \quad (1)$$

Al finalizar, el nodo “*extrinsics_from_plane*” publica los parámetros a , b , c y d de (1), además de los ángulos pitch y roll. Estos parámetros serán recogidos en el siguiente nodo para construir la vista de pájaro.

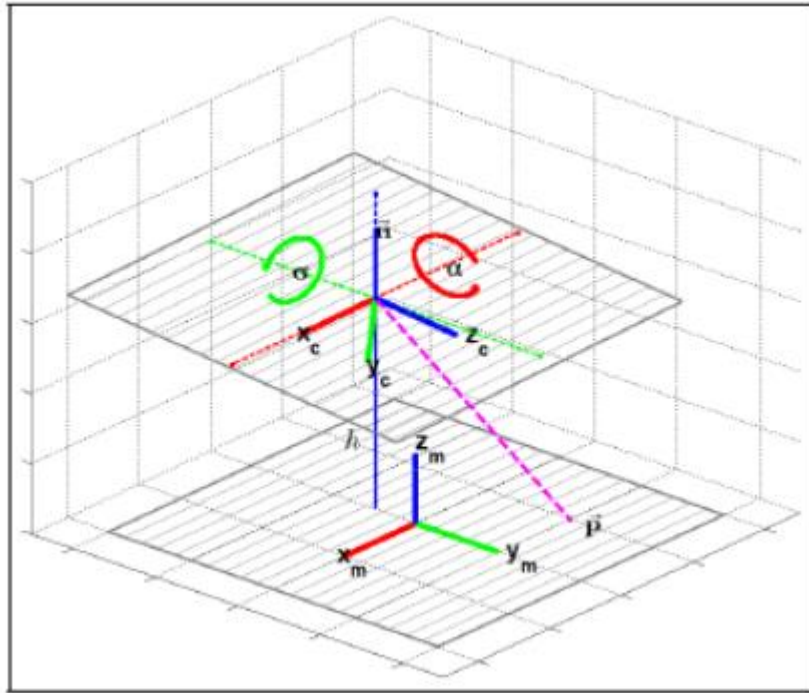


Figura 20: Representación de los parámetros extraídos del plano de la carretera

4.1.4. BIRD VIEW (Vista de pájaro):

Una vez identificado el plano de la carretera, a través del siguiente nodo, las imágenes pueden ser proyectadas a la vista superior o vista de pájaro. El objetivo de realizar esta transformación es la de poder identificar más fácilmente el carril a través de una de las principales características de estos, el paralelismo, ya que en la vista de pájaro el paralelismo es conservado visualmente, Figura 21, al contrario que ocurre en una imagen sin tratar, debido a la perspectiva de la carretera.

Tal y como se explica en [6], este cambio de perspectiva se efectúa mediante la transformación homográfica:

$$s^c P = {}^c P_s = {}^c H_{vp} {}^{vp} P \quad (2)$$

Además, de esta manera, es posible, modificando la posición de la cámara, cambiar la región de interés mostrada en la imagen (ROI_{vp}), pudiendo desplazar la escena horizontal o verticalmente.

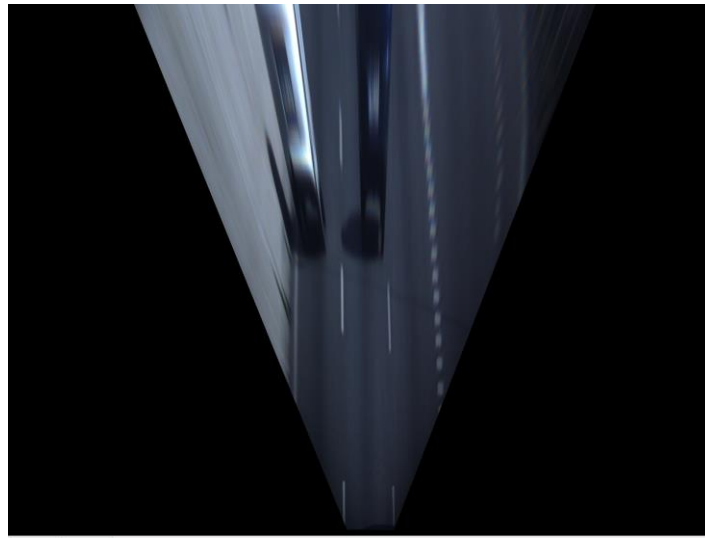


Figura 21: Vista de pájaro (Bird View)

Dado que las etapas siguientes son aplicadas en la perspectiva vista de pájaro está es publicada en el tópico “/stereo_camera/bird_view”, además se muestra en una ventana independiente para poder monitorizar el correcto funcionamiento de manera cómoda.

4.1.5. LANE DETECTION:

Se trata del nodo principal y a su vez del más extenso ya que es el encargado de reconocer el carril, así como de clasificarlo y remarcarlo en la calzada. Consta de una serie de etapas o procesos necesarios para conseguir los

mejores resultados. La metodología y algoritmos utilizados están extraídos del libro [27].

Máscara de Marcas Viales:

El primer proceso a realizar es, dado que en la vista de pájaro quedan reflejados muchos objetos considerados como ruido (coches, señales, etc.), filtrar la imagen a través de máscaras de marcas viales en el espacio del gradiente. Gracias a esta máscara, al simplificar la imagen se consigue reducir el espacio de aplicación, así como filtrar la imagen teniendo en cuenta las características geométricas que tienen las marcas viales, tales como el ancho de una línea separadora de carril o la diferencia de color entre la línea y el fondo donde está pintada.

En una primera etapa se detectan los bordes de la imagen. Para esto se utilizan los tres primeros pasos del algoritmo Canny [7] con algunas modificaciones en estos pasos, tales como el uso de un estimador de gradiente distinto que el algoritmo original propuesto. La forma de operar es básicamente la aplicación de un filtro por convolución tipo campana de Gauss para difuminar la imagen. Posteriormente se eliminan los bordes débiles (píxeles con valor de gradiente inferior a un umbral escogido previamente, de tal forma que la mayoría de bordes sean conservados.

En la segunda fase, cuando ya se tienen los bordes de la imagen localizados gracias al detector de Canny, se buscan pares de puntos que puedan ser de tipo borde, es decir, que formen los extremos de cada segmento de una marca vial. Dichos segmentos, a su vez, están formados por una serie de puntos sucesivos cuyos gradientes deben de cumplir unas restricciones (3) en la posición inicial y final. Estas restricciones son:

$$\begin{aligned} 1. & \left| \|\nabla p_{j,1}\| - \|\nabla p_{j,n(v_j)}\| \right| < \epsilon_m, \\ 2. & \left| \left| \angle \nabla p_{j,1} - \angle \nabla p_{j,n(v_j)} \right| - \pi \right| < \epsilon_a, \\ 3. & \left| \|p_{j,1} - p_{j,n(v_j)}\| - w_{mv} \right| < \epsilon_w \end{aligned} \quad (3)$$

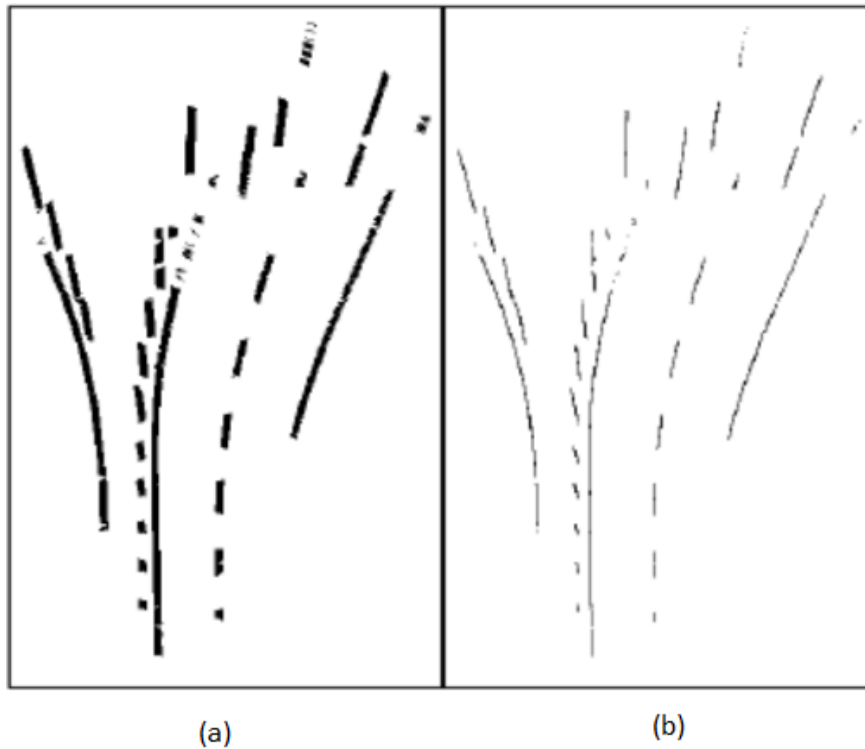
1. La magnitud de los gradientes debe ser aproximadamente iguales en la posición inicial y final.

2. La dirección del gradiente en cada uno de los extremos debe ser la opuesta al otro.

3. La longitud del segmento debe ser cercana al ancho de una marca vial. Este ancho de vía es fijado correspondientemente con la altura de la cámara virtual en la vista de pájaro.

De esta manera, gracias a la diferenciación entre marca vial y fondo, se eliminan los bordes débiles de la escena.

Por último, en una tercera fase, se construye la máscara de marcas viales, **RM**, mediante la superposición de los segmentos detectados como parte de marcas viales. Además, para extraer el centro de las marcas viales se realiza un análisis sobre cada segmento detectado en el cual se toma como hipótesis de partida que cada segmento tiene un sub-segmento de magnitud mínima correspondiente a la zona blanca de la marca vial. En la Figura 22 se muestra la máscara de marcas viales (a) y la extracción de las líneas centrales del ancho del carril (b).



*Figura 22: (a) Máscara de Marcas Viales; (b) Extracción de las líneas
centrales del ancho del carril*

Detección de Líneas y Carriles Viales:

Una vez construida la máscara de marcas se comienza con el proceso de detección de líneas, mediante el uso del algoritmo o transformada de Hough [8]. Como se puede ver en la Figura 23 son detectadas multitud de líneas tanto de la carretera como otras líneas sin importancia.

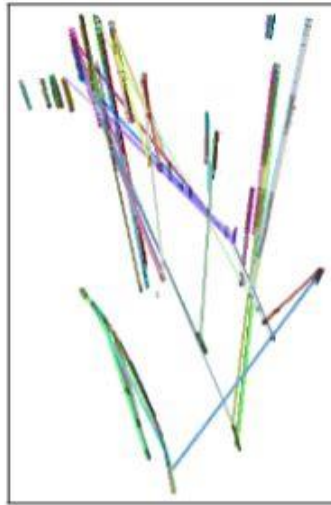


Figura 23: Líneas detectadas al realizar la transformada de Hough

Ya que se han detectado líneas que no son de interés, es decir, que no pertenecen a marcas viales y que por tanto han de ser desechadas, es necesario filtrar o clasificar estas líneas. Esta clasificación se realiza filtrando pares de líneas paralelas entre sí y con una separación igual a la longitud transversal de carril estándar, la cual viene reflejada en la norma IC 8.2 [9]. Sin embargo, estas características no siempre filtran correctamente las líneas de la carretera, dado que puede darse el caso en el que un par de líneas cumplan dichas condiciones sin llegar a pertenecer a la carretera. Para terminar de eliminar estos falsos positivos, es necesario conocer el ángulo de la carretera, y de esta manera sí que es posible filtrar correctamente las líneas que realmente pertenecen a una marca vial.

A través de unos procesos matemáticos [15], se extrae el ángulo de todas las líneas detectadas, y se consigue extraer el ángulo predominante de todos ellos. Este ángulo será identificado como la dirección de los carriles a lo largo de la carretera.

Una vez que se conoce el ángulo de todos los pares de líneas detectados, se construye un conjunto de carriles X , donde cada carril X_i cumple con las

condiciones de tener dos líneas paralelas entre sí, tener el ángulo identificado como predominante, y tener una separación la una de la otra igual al ancho de carril estipulado. De este conjunto X se extrae un subconjunto con mayor número de elementos y que además cumple con la condición de ser adyacentes y no solapados. A este subconjunto se denominará como *estructura de carretera*, R .

Clasificación de los carriles:

La clasificación de los carriles es, junto con otras etapas como la detección de líneas, una etapa muy importante para el correcto entendimiento de la carretera para poder ejecutar acciones autónomas como adelantamientos, cambios de carril o toma de decisiones en función del tipo de carril. Así, conociendo el tipo de carril por el que se está circulando o los adyacentes, es posible mejorar considerablemente procesos como seguimiento de carriles u odometría visual.

Para comenzar con la clasificación de carriles, se introduce al clasificador un perfil binario P_i (5), el cual es extraído de la máscara de marcas viales RM (4). Dicho perfil está compuesto por una colección de puntos pertenecientes a la estructura de carretera, R . Estos puntos definen los extremos de cada línea detectada sobre la RM .

$$P_i(t) = RM(l_i(t)) \quad (4)$$

$$l_i(t) = (long(l_i) - t)P0_i + tP1_i \quad (5)$$

Donde $i = 1, 2, \dots, n^{\circ}líneas$, $t = 1, 2, \dots, long(l_i)$, $P0_i$ y $tP1_i$ son el punto inicial y final de la línea respectivamente.

Se han propuesto dos tipos de clasificación de carriles:

- Construcción manual del modelo:

En esta opción, es necesario tener un modelo con puntos representativos de cada tipo de carril. Este modelo es construido manualmente a partir de una colección de seis mil líneas detectadas. De esta manera se obtienen dos gráficas representadas en la Figura 24, en (a) se observa la gráfica construida a partir del valor medio de líneas frente a la longitud de línea, y en (b) se representa el resultado de representar la frecuencia primer pico de potencia frente frecuencia segundo pico de potencia. Una vez construida la gráfica se asignaron colores a cada punto en función de su tipo de carril, así el rosa identifica una línea continua, el azul una línea discontinua, el rojo una línea de incorporación y por último el verde una línea desconocida.

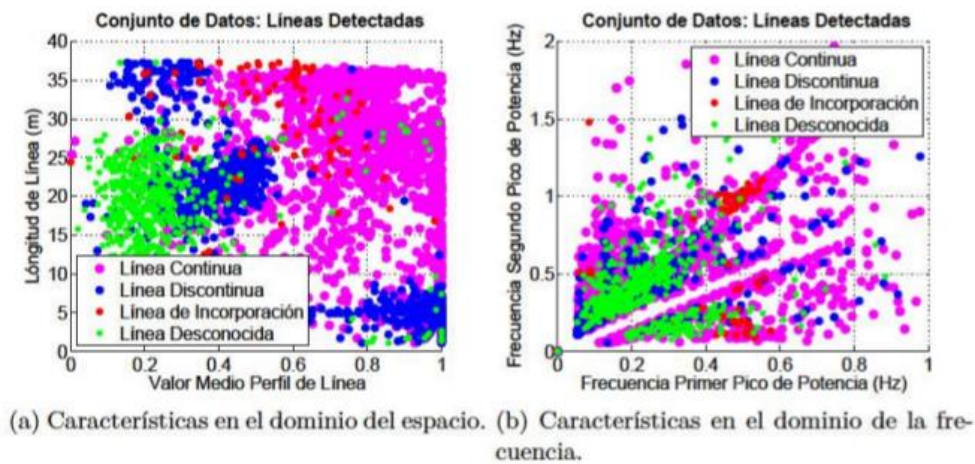


Figura 24: Modelo introducido de manera manual para la clasificación de carriles

Una vez construido el modelo, lo único que hay que hacer es comparar cada línea con dicho modelo para identificar de qué tipo de línea se trata.

- Modelo a partir de Maquinas de Soporte Vectorial [31]:

Con esta opción el modelo a entrenar se extrae directamente de una Máquina de Soporte Vectorial (SVM), [20]. Estas máquinas son herramientas fundamentales en sistemas de aprendizaje automático, permitiendo el tratamiento de problemas actuales en reconocimiento de patrones y minería de datos. Las SVM tienen un rol muy

importante en teoría de aprendizaje estadístico y cuando es necesario entrenar un clasificador, como es en nuestro caso.

En nuestro caso esta SVM cargará un modelo similar al introducido manualmente en la opción anterior, aunque con una clasificación mejorada.

La elección del tipo de clasificador está a disposición del usuario a través del parámetro “*lane/line_classifier*” el cual puede tomar valor 0, clasificador mediante el modelo introducido manualmente, o 2, clasificador basado en un entrenador SVM. Dicho parámetro puede ser introducido en el *.launch* con nombre “*tfg_launch.launch*” o bien introduciendo el parámetro directamente por la terminal, mediante el comando “*roscparam set ...*”.

De manera general y si el usuario no introduce ningún valor para el parámetro anterior se tomará dicho valor a 2 y por tanto se utilizará un clasificador SVM ya que los resultados obtenidos son mejores mediante el uso de este frente al resto de clasificadores propuestos.

En la Figura 25 se muestra un esquema por imágenes del resultado de las principales operaciones explicadas anteriormente.

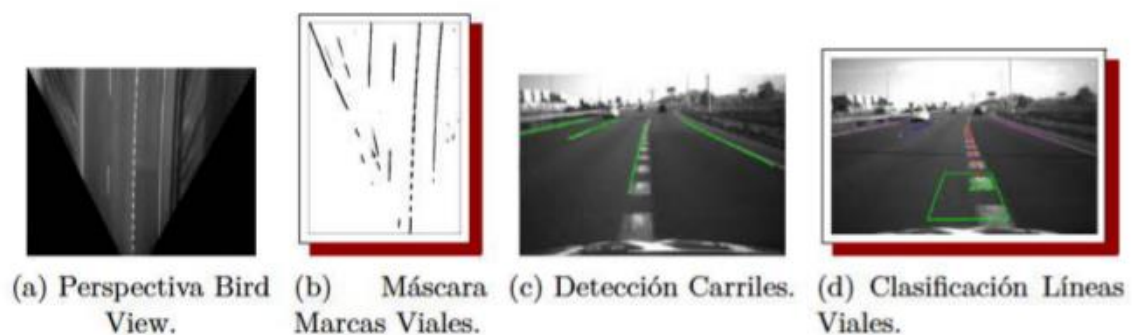


Figura 25: Esquema por imágenes de los pasos llevados a cabo para la detección de carril

CAPÍTULO 5: INTEGRACIÓN Y ACTUALIZACIÓN DE LOS NODOS

Si bien se ha explicado en los capítulos anteriores la necesidad de realizar determinadas acciones para la correcta detección y clasificación de las marcas viales, tales como calcular un mapa de disparidad, creación de la PC, extracción del plano, etc., estos procesos se realizaban en un mismo programa, lo que no permitía la reutilización de código para otras aplicaciones ADAS. Es por esto que se decidió cambiar la arquitectura del IVVI 2.0.

En este TFG se han reutilizado distintos paquetes de ROS, un programa para el cálculo del mapa de disparidad, otro programa para la creación de la nube de puntos, otro para la extracción y calibración del plano, uno para el cambio de perspectiva a vista de pájaro, y por último el programa con el resto de métodos necesarios para la detección y clasificación, y se han integrado todos ellos para el correcto funcionamiento del programa.

Con la configuración actual, si otro sistema ADAS implementado en el IVVI 2.0 necesita, por ejemplo, la utilización de la PC o de la *Bird View*, solo tendrá que subscribirse al tópico correspondiente y ahorrarse esos pasos, reduciendo así el tiempo de desarrollo del programa y aumentando la velocidad de computación.

A continuación, se expone un resumen de los tópicos y de los cambios más significativos que han sufrido los nodos más importantes:

Bird View:

Una vez obtenida la vista de pájaro tal y como se explica en el Capítulo 3 se creará una nueva ventana en la que se publicará dicha vista como formato Mat gracias a las OpenCv explicadas en el punto 3 del capítulo anterior. Además, dicha variable Mat es publicada en el tópico “/stereo_camera/bird_view”. Este nodo ha sido modificado durante la realización de este trabajo de tal manera que la extracción de variables internas, necesarias para la ejecución del código, se realiza directamente desde el tópico “/stereo_camera/left/camera_info” y con la consiguiente mejora de la correcta funcionalidad de dicho nodo para diferentes tamaños de imagen (640x480, 1024x768 y 1280x960).

Estos parámetros son: *nrows* (número de filas de la imagen a procesar), *ncols* (número de columnas), *fx* y *fy* (distancias focales) y *cx* y *cy* (puntos principales). Estos parámetros son necesarios para cálculos posteriores, y la obtención de estos se realiza mediante sentencias definidas. En la Figura 26 se muestra un ejemplo de sentencia para la obtención de los parámetros *nrows* y *ncols*, y en la referencia [33] se muestra la documentación del tópico “*sensor_msgs/CameraInfo*”

```
nrows=cam_info->height;  
ncols=cam_info->width;
```

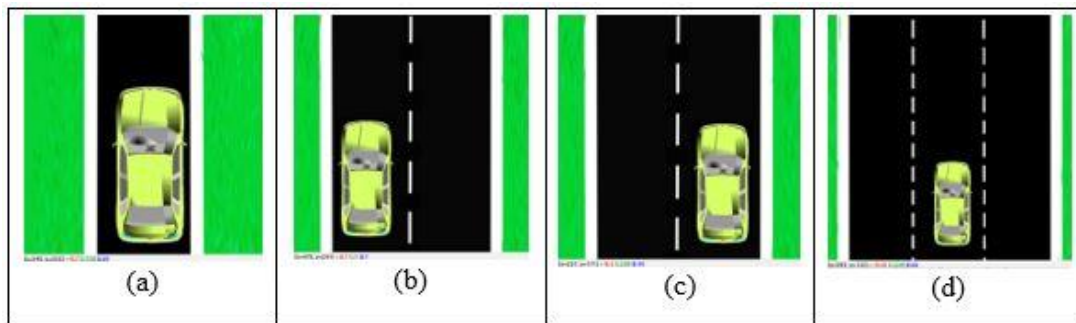
*Figura 26: Métodos utilizados en el código para la obtención de las variables *nrows* y *ncols**

Posicionamiento del vehículo:

Gracias a una serie de algoritmos, desarrollados en [24] es posible calcular la posición relativa del coche en relación a la anchura del carril detectado y, por tanto, conocer la distancia del vehículo a una línea divisoria, con lo que se podría corregir la dirección del coche en caso de que este se encuentre en situación de sufrir una salida de carril.

Como se ha explicado en el Capítulo 4, es posible conocer el número de carriles que hay en la vía por la que circula el coche, así como el carril exacto de esa vía que está siendo ocupado por dicho vehículo.

Con estos dos datos es posible mostrar, en una ventana, una representación de la carretera detectada situando el coche en el carril exacto y dentro de dicho carril la posición relativa que ocupa en cada instante tal y como se ve en la Figura 27. Dicha ventana es muy útil mostrarla tanto para monitorizar los resultados, como para ser mostrada en el salpicadero del coche para ofertar dicha información al usuario del vehículo. Además, esta imagen es publicada en el tópico “*/lane_detection/carretera*” con lo que puede ser utilizada en trabajos futuros.



*Figura 27: Representación de posibles situaciones de carreteras, así
como posicionamiento relativo del vehículo dentro del carril*

Resultados finales:

Por último, también se muestra en una ventana la imagen capturada por la cámara con los carriles detectados dibujados con sus respectivos colores en función del tipo de carril, tal y como se explica en el capítulo 4. Esta información también es publicada en el tópic `“/lane_detection/adas”`.

Para poder dibujar dichos carriles con los distintos colores es necesario realizar el proceso inverso al realizar la vista de pájaro. Para esto se toman los puntos de inicio y final de carril, así como el tipo de carril y al hacer la transformada inversa volviendo a la vista estéreo se obtienen las proyecciones originales de dichos puntos pintando el carril del color correspondiente. Dichos resultados se pueden ver en la Figura 28. En 31(a) se muestran los resultados junto con una brújula en la parte inferior de la imagen, la cual representa la posición y el ángulo del coche en el carril. En 31(b) se muestran dichos resultados, pero sin la brújula mencionada anteriormente, ya que esta se representa en la imagen de la carretera. Para elegir si se desea o no la visualización de esta brújula se puede hacer a través del `.launch` `“tf.launch.launch”` dando valor `“true”` o `“false”` al parámetro `“lane/show_pos”`. Dicho parámetro también se puede introducir a través de la terminal.

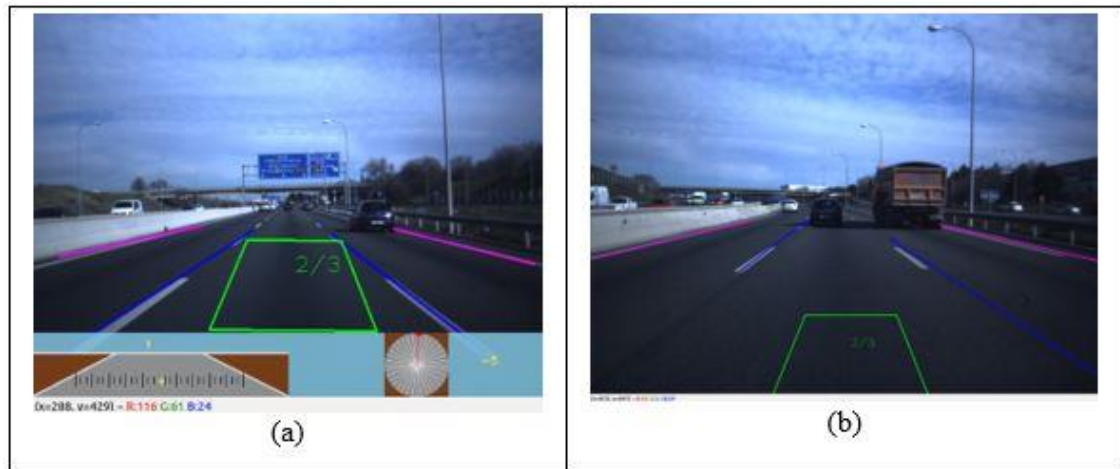


Figura 28: Visualización de los dos tipos de vistas

Dado que se ha posibilitado la funcionalidad del nodo *Bird View* para diferentes tamaños, ha sido necesario la actualización de la obtención de los parámetros intrínsecos de la cámara, de tal manera que también son extraídos de manera similar al nodo anterior.

Otras actualizaciones realizadas en este nodo son:

- * Actualización del código para el correcto funcionamiento del filtro Kalman, el cual es utilizado para suavizar los saltos de valor en la posición del vehículo en el carril que se producen entre distintos *frames* del video.

- * Actualización a OpenCV 3 del código para el correcto funcionamiento del clasificador basado en Maquinas de Soporte Vectorial (SVM).

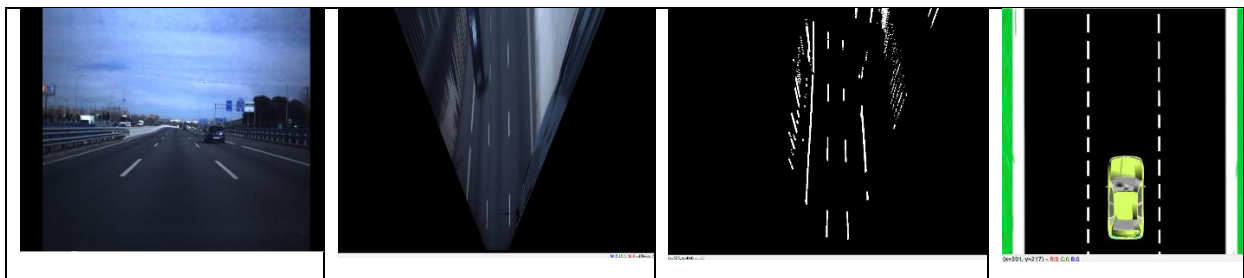
- * Se ha añadido un nuevo nodo con el fin de poder redimensionar la imagen entrante al tamaño deseado por el usuario. De esta manera se puede seleccionar dicho tamaño introduciendo los valores del tamaño de pantalla buscado en el *launch* llamado "*scale_stereo.launch*".

* Además se han hecho otras mejoras de menor magnitud tales como posibilitar la introducción al programa de imágenes a color, ya que la implementación antigua no permitía este formato, o la utilización de operadores matemáticos, tales como senos o cosenos, a través de las librerías de C++ en lugar de a través de las librerías de OpenCV, ya que esta generaba errores debidos al cambio a la nueva versión OpenCV 3.

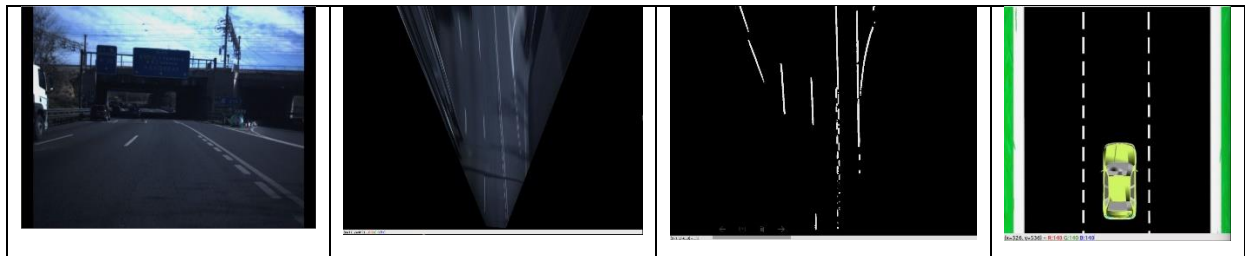
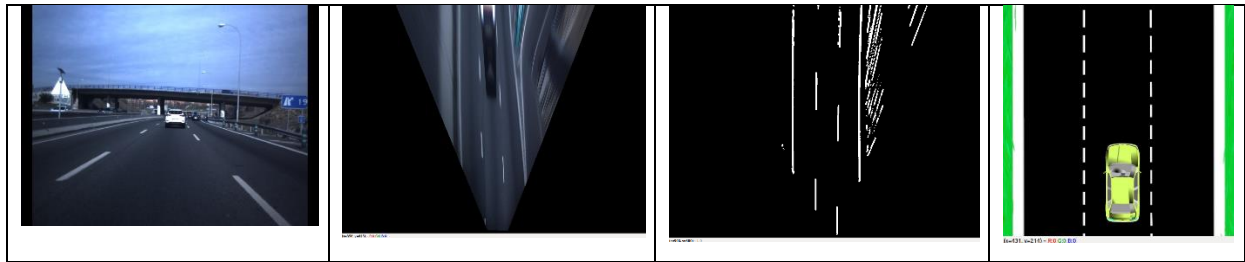
CAPÍTULO 6: RESULTADOS OBTENIDOS

Con el fin de comprobar la fiabilidad del trabajo realizado, se han llevado a cabo una serie de pruebas. La finalidad de este programa es la de ser implementado en el IVVI 2.0 y realizar el proceso en tiempo real, es decir, que la entrada al programa venga de las cámaras estéreo situadas en el frontal del vehículo. Sin embargo, para la comprobación del funcionamiento, se han grabado una serie de videos con dichas cámaras para simular esta situación. Dichas grabaciones son convertidas a un archivo *.bag*, el cual es el formato utilizado para reproducir en ROS dos imágenes estéreo al mismo tiempo. Estas imágenes, al cargar el *.bag* correspondiente, son publicadas en el tópic *"/stereo_camera/address/image_rect_color"*, donde *"address"* será *"left"* para la imagen capturada por la cámara izquierda, y *"right"* para la imagen capturada por la cámara derecha.

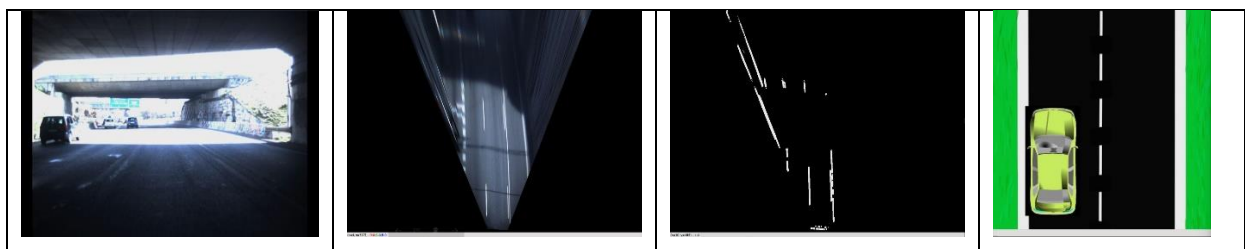
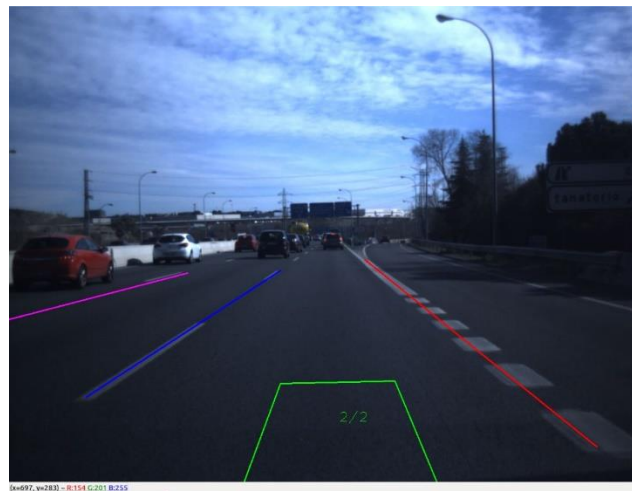
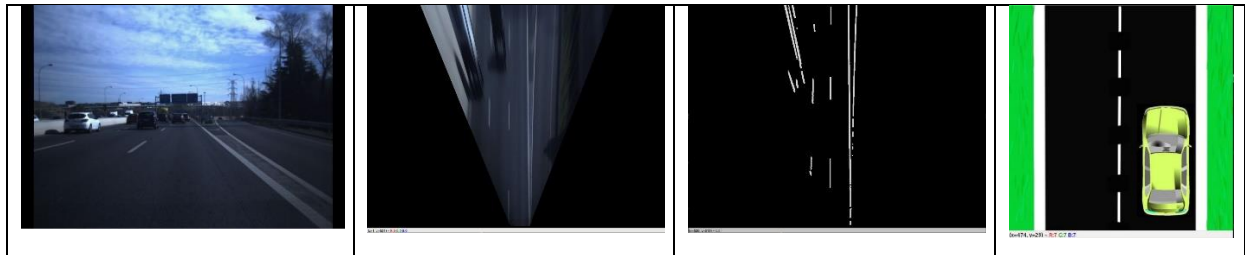
Se han reproducido distintos *.bag* para comprobar el funcionamiento del programa en diferentes situaciones de circulación reales. Los resultados obtenidos han sido los siguientes:



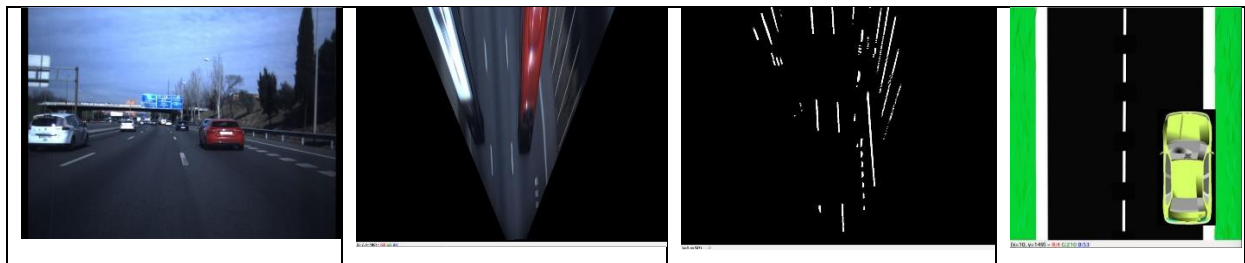
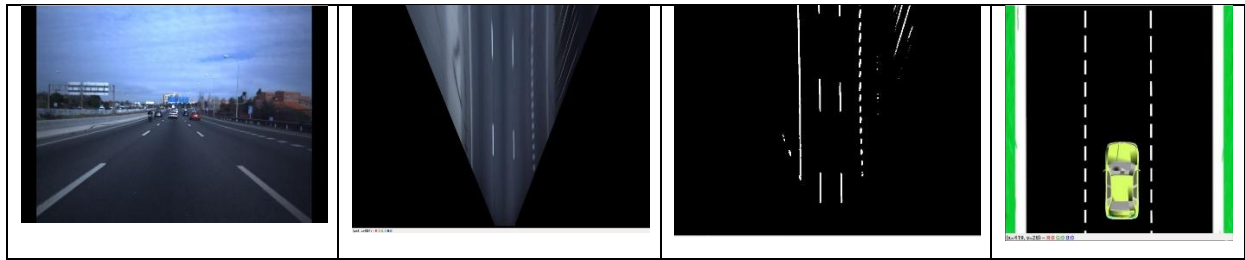
Implementación de algoritmo de modelado de carriles en la arquitectura de software del vehículo inteligente IVVI 2.0



Implementación de algoritmo de modelado de carriles en la arquitectura de software del vehículo inteligente IVVI 2.0



*Implementación de algoritmo de modelado de carriles en la arquitectura de software del vehículo
inteligente IVVI 2.0*





CAPÍTULO 7: MARCO REGULADOR

En los últimos años la industria automovilística ha sufrido un cambio muy brusco, tal y como se ha expuesto en el Capítulo 1, con una evolución importante en los sistemas denominados ADAS. Es tal el avance conseguido en estos sistemas, que la creación de un coche 100% autónomo parece estar cada vez más cerca.

Esta rapidez con la que está avanzando la tecnología del coche autónomo ha llevado a las diferentes instituciones gubernamentales de diferentes países, a la elaboración de un marco regulador en esta materia.

En España, las primeras pruebas en cuanto a regulación aparecieron a finales de 2015, cuando la DGT aprobó una primera normativa que permitía la prueba de vehículos autónomos por las carreteras españolas, tal y como se cuenta en [21]. En esta primera regulación se especificaba que el uso de coches autónomos por las vías de circulación, únicamente podía realizarse para la investigación y desarrollo de las nuevas tecnologías.

Actualmente, como se expone en [22], la DGT, desde el proyecto de Movilidad Autónoma y Conectada, ya trabaja en un primer reglamento oficial que regule la circulación y características de este tipo de vehículos. En este reglamento, según [22], se incluirán capítulos y anexos dedicados a la conducción automatizada, así como se hablará de definiciones, requisitos, y niveles de automatización de los vehículos para dar un soporte jurídico. Según [22], la aprobación de esta regulación no se llevará a cabo hasta 2018 o 2019.

Sin embargo, aunque no exista una regulación clara de la utilización del coche autónomo en España, es posible que un particular circule en España con un vehículo que se conduzca solo, ya que España, al no adherirse a la convención de Viena firmada en 1968, no está sujeta a ciertas limitaciones que si tienen otros países europeos. Es por esto, que ya se pueden ver circulando Teslas con el modo Autopilot en nuestro país, a pesar de no existir una regulación oficial.

En cuanto al marco regulador en otros países, cabe destacar el de Estados Unidos, por ser el país con mayor número de coches autónomos. Como se cuenta en [23], ya existe un marco legal para los coches autónomos a nivel federal.

Este proyecto de ley, el cual se estima que podría ser aprobado antes de finalizar el 2017, prohibiría a los distintos Estados regular los sistemas mecánicos, de software y de seguridad de estos automóviles, quedando esta función en manos del gobierno federal.

CAPÍTULO 8: ENTORNO SOCIOECONÓMICO

8.1. PRESUPUESTO

Observando las tablas de amortizaciones publicadas este año 2017 por la Agencia Tributaria del Gobierno de España, [24], se extrae que para los siguientes elementos necesarios para el desarrollo de este trabajo, el coeficiente lineal máximo es: equipos electrónicos, se tiene una amortización del 20%; sistemas y programas informáticos, una amortización del 33%, aunque dado que los programas informáticos son de software libre, el gasto por amortización de estos bienes serán nulos.

El cálculo de la amortización de los elementos utilizados durante la realización del trabajo se realiza mediante la fórmula mostrada en (6), donde *coef.amort* es el porcentaje de amortización expuesto en el párrafo anterior dividido entre 100, y *t* es el tiempo de uso de este (en años).

$$\text{Amortización} = \text{Precio Adquisición} * \text{coef.amort} * t \quad (6)$$

El tiempo *t* considerado ha sido de 8 meses. De esta manera se obtiene la siguiente tabla:

Elemento	Precio de Compra (€)	Coeficiente de amortización	Amortización
HP Omen 15-ax001ns	1060	0,2	141,33

Tabla 3: Amortización del equipo utilizado durante la realización del trabajo.

Para la realización de este trabajo fue contratado un ingeniero, suponiendo un salario de 1.800 €/mes, supondrá un gasto total de 14.400 €/mes.

Con lo que el Coste Total del proyecto será de:

Concepto	Importe	
Coste material	141,33	
Coste personal	14400	
	TOTAL	14541,33

Tabla 4: Estimación de Gastos Totales

8.2. IMPACTO SOCIOECONÓMICO:

El vehículo autónomo, y todas las aplicaciones que con este se presentan, transformará nuestra sociedad y economía, llevándola a un punto muy diferente del actual [38]. Pudiendo producirse diferentes variaciones de mercado, algunas muy difundidas y obvias y otras variaciones que pasan más desapercibidas y son igualmente muy importantes.

El primer posible cambio es el referido a la conducción eficiente y a un descenso del consumo energético. Además el número de conductores aumentará, ya que se expandirá la banda de edad a partir de la que se permitirá utilizar uno de estos vehículos sin acompañantes adultos. El sector de la logística y el transporte también sufrirá un cambio importante, donde ya se están haciendo pruebas con barcos autopilotados.

Pero si hablamos de cambios importantes, hay que mencionar el cambio que sufriría el sector de los seguros, ya que las cuestiones que se plantean en este campo afectan a determinar quién sería el responsable subsidiario de un accidente, cuando un vehículo autónomo esté involucrado.

Uno de los puntos de vista socioeconómicos más interesante es que los vehículos autónomos formarían una flota inmensa de coches circulando de manera masiva por las ciudades, dispositivos con una inteligencia admirable con capacidad para portar sensores y medios muy superiores a nuestros móviles, por ejemplo. Esto haría que involuntariamente, con nuestros trayectos, fuésemos poblando el mapa de fotos, información meteorológica y estado de las carreteras, accidentes, atascos, etc.

Pero sin duda el cambio socioeconómico más importante se dará en la industria del automóvil, ya que las empresas correspondientes transformarán dicho sector a un nivel superior al del vehículo eléctrico.

CAPÍTULO 9: CONCLUSIÓN Y TRABAJOS FUTUROS

9.1. CONCLUSIONES

En relación a los resultados esperados al implementar el programa de detección de marcas viales en la nueva arquitectura del IVVI 2.0 basada en ROS, han sido positivos en todos los aspectos, ya que no solo se ha mantenido la calidad de dicho programa, sino que se han añadido mejoras tales como la posibilidad de funcionar en distintas escalas de imagen, así como mostrar la posición relativa del vehículo en la carretera de manera más visual, o publicar todos los resultados en los correspondientes tópicos y con ello las facilidades tanto para la reutilización de código en dicho programa como las facilidades para futuras funciones que se quieran implementar en el vehículo IVVI 2.0 tales como reconocimiento de semáforos, señales o cualquier propuesta relacionada con los sistemas ADAS.

En cuanto a complicaciones encontradas, se han encontrado algunas dificultades para conseguir un correcto funcionamiento de las funciones actualizadas a la versión OpenCV 3, haciendo un especial hincapié en el clasificador SVM y en el filtro Kalman.

9.2. TRABAJOS FUTUROS

Durante la realización de este trabajo han aparecido distintos problemas, algunos de los cuales se han podido solucionar, mientras que otros no. Estos problemas han abierto una serie de campos de mejora. Algunas de estas son:

- Mejorar el modelo de la carretera para zonas con pérdidas de información, de tal manera que el nuevo modelo sea capaz de contemplar determinadas situaciones, tales como, el adelantamiento de camiones o vehículos de grandes dimensiones, lo que provoca la desaparición momentánea de un carril, ya que el tamaño de estos, impide la visualización de la línea contigua a este. Otros ejemplos de factores que pueden generar perdidas momentáneas de información son, el paso por zonas de sombra, el paso por pequeños túneles no iluminados o puentes, o condiciones de baja visibilidad, como lluvia.

- Incluir en el modelo de carretera actual la posibilidad de reconocer carriles en curvas pronunciadas, situación en la que actualmente, es muy difícil para el sistema reconocer correctamente las marcas viales.
- Plantearse ampliar el rango de tipos de carriles a detectar. Sobre todo, añadir la posibilidad de reconocer líneas dobles, presentes en carreteras convencionales.
- Otra mejora importante, y que mejoraría considerablemente los resultados obtenidos, sería la de combinar la información obtenida desde el par estéreo, con información obtenida de sensores.
- Mejora del algoritmo propuesto, de tal manera que en caso de pérdida momentánea de algún carril, el sistema sea capaz de, mediante el conteo de *frames* sin detección, decidir si el carril realmente ha desaparecido o por el contrario se trata de un error de percepción, con lo que sería capaz de mantener el carril perdido como detectado.
- Por último, se propone combinar este sistema con algún sistema de posicionamiento GPS, de tal manera que el vehículo sepa en que tipo de vía se encuentra, y en que posición está respecto a esta, pudiendo partir con el conocimiento previo del número de carriles que forman la vía. Pudiendo así, una vez ubicada la posición del vehículo en relación al ancho de vía, ser capaz de construir un modelo a lo largo del carril.

CAPÍTULO 10: BIBLIOGRAFÍA

- [1] M. Bordacheva, «Aportación de la industria del automóvil al crecimiento económico de España e impacto de las tendencias actuales en el sector,» 2015.
- [2] D. G. d. T. (DGT), «Las principales cifras siniestralidad 2015,» 2015.
- [3] P. G. Bejerano, «La desconocida historia de los coches autónomos,» 12 agosto 2013. [En línea]. Available: <https://blogthinkbig.com/historia-de-los-coches-autonomos>. [Último acceso: septiembre 2017].
- [4] «Vehículo autónomo,» [En línea]. Available: https://es.wikipedia.org/wiki/Veh%C3%ADculo_aut%C3%B3nomo. [Último acceso: septiembre 2017].
- [5] D. Martín Gómez, «IVVI 2.0: An intelligent vehicle based on computational perception,» 2014.
- [6] «Asistente para visión nocturna Plus,» 2017. [En línea]. Available: http://techcenter.mercedes-benz.com/es_ES/night_view_plus_with_spotlight/detail.html. [Último acceso: septiembre 2017].
- [7] «Asistente de visión nocturna con luz de marcado de peatones y animales,» 2017. [En línea]. Available: <http://www.audi.es/es/web/es/innovacion-audi/tecnologia/asistente-de-vision-nocturna.html>. [Último acceso: septiembre 2017].
- [8] «El sistema ACC de Bosch regula la velocidad del coche, manteniendo la distancia de seguridad,» 2013. [En línea]. Available: <http://www.posventa.info/tecnica-y-tecnologia/el-sistema-acc-de-bosch-regula-la-velocidad-del-coche-manteniendo-la-distancia-de-seguridad/3715/>. [Último acceso: septiembre 2017].

- [9] «Sistemas de asistencia a la conducción,» 2017. [En línea]. Available: http://www.bosch-mobility-solutions.es/es/es/driving_safety_8/driving_safety_systems_for_passenger_cars_9/driver_assistance_systems_32/driver_assistance_systems_2.html#. [Último acceso: septiembre 2017].
- [10] J. M. P. Rastelli, «Agentes de control de vehículos autónomos en entornos,» 2012.
- [11] S. V. S. M. Cartuche Minga Segundo Pachacutic, «Diseño, construcción e implementación de un sistema de reconocimiento de objetos en carretera durante la conducción, bajo condiciones adversas, en un vehículo,» 2015.
- [12] «Asistente de mantenimiento en carril: ¿comodidad o seguridad?,» 2013. [En línea]. Available: <http://www.circulaseguro.com/asistente-de-mantenimiento-en-carril-comodidad-o-seguridad/>.
- [13] W. b. C. Team, «LDWS y BSD: las tecnologías que te protegen de los cambios de carril involuntarios y los ángulos muertos,» 2015. [En línea]. Available: <https://www.xataka.com/tecnologiakia/ldws-y-bsd-las-tecnologias-que-te-protegen-de-los-cambios-de-carril-involuntarios-y-los-angulos-muertos>. [Último acceso: septiembre 2017].
- [14] J. M. C. Hernáiz, «Detección y Modelado de Carriles de Vías Interurbanas mediante Análisis de Imágenes para un Sistema de Ayuda a la Conducción,» 2009.
- [15] A. J. Hawkins, «Meet ALVINN, the self-driving car from 1989,» 2016. [En línea]. Available: <https://www.theverge.com/2016/11/27/13752344/alvin-self-driving-car-1989-cmu-navlab>. [Último acceso: septiembre 2017].
- [16] D. Martín de Consuegra Martínez, «Diseño y desarrollo de

- una interfaz gráfica de usuario para la prueba de DAQS basados en Arduino mediante ROS,» 2012.
- [17] A. R. Bazaga, «OpenCV: Librería de Visión por Computador,» 2015. [En línea]. Available: <https://osl.ull.es/software-libre/opencv-libreria-vision-computador/>. [Último acceso: septiembre 2017].
- [18] F. J. P. Pérez, «Herramientas de la Librería PCL,» 2013. [En línea]. Available: <http://pointcloudlibrary.blogspot.com.es/2013/12/herramientas-de-la-libreria-pcl.html>. [Último acceso: septiembre 2017].
- [19] S. Chacon y B. Straub, Pro Git, Apress, 2014.
- [20] J. S. Garde, «Cálculo de mapas de disparidad mediante block matching,» 2014. [En línea]. Available: <http://losordenadorestienenojos.blogspot.com.es/2014/06/calculo-de-mapa-de-disparidad-mediante.html>.
- [21] pcl, «Downsampling a PointCloud using a VoxelGrid filter,» [En línea]. Available: http://pointclouds.org/documentation/tutorials/voxel_grid.php. [Último acceso: septiembre 2017].
- [22] P. J. Rousseeuw, «Least Median of Squares Regression,» *Journal of the American statistical association*, 1984.
- [23] D. G. Aguilera, «PROCESAMIENTO DE IMÁGENES».
- [24] C. H. Rodríguez Garavito, «Sistema avanzado de asistencia a la conducción para entornos interurbanos,» 2017.
- [25] C. P. A. G. F. M. D. d. I. E. A. A. J. Rodríguez Garavito, «Automatic laser and camera extrinsic calibration for data fusion using road plane. In Information Fusion (FUSION),» *International Conference On. IEEE*, 2014.
- [26] G. A. Rodríguez Garavito, «Sistema de Asistencia a la Conducción para Detección y Clasificación de Carriles,» 2015.

- [27] C. H. C.-F. J. d. I. E. A. & A. J. M. Rodríguez-Garavito, Stereo Road Detection Based on Ground Plane, Springer, 2015.
- [28] J. Valverde-Rebaza, «Detección de bordes mediante el algoritmo de Canny,» 2007.
- [29] J. L.-M. L. N.-D. Luis Canul-Arceo, «Algoritmo rápido de la transformada de Hough,» 2015.
- [30] M. d. O. P. y. Urbanismo, «Norma de carreteras 8.2-IC. Marcas Viales,» 1987.
- [31] C. H. P. A. & G. F. Rodríguez-Garavito, COMPUTER VISION APPLIED TO ROAD LINES RECOGNITION USING MACHINE LEARNING, ICIT, 2015.
- [32] T. T. V. A. A. D. G. Javier Giacomantone, «Reconocimiento Estadístico de Patrones».
- [33] ROS, «sensor_msgs/CameraInfo Message,» 14 Jun 2017. [En línea]. Available:
[http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.htm](http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html)
l. [Último acceso: septiembre 2017].
- [34] c. GONZÁLEZ, «Los coches autónomos ya son legales en España,» 2015. [En línea]. Available:
<https://testcoches.es/actualidad/los-coches-autonomos-ya-son-legales-en-espana/>. [Último acceso: septiembre 2017].
- [35] J. Esteve, «España se pone las pilas: la DGT regulará por primera vez el coche autónomo en 2017,» 2016. [En línea]. Available: https://www.elconfidencial.com/tecnologia/2016-12-23/coche-autonomo-espana-dgt-2017-reglamento_1308238/. [Último acceso: septiembre 2017].
- [36] Y. R. Fernandez, «Aprobado un marco legal para los coches autónomos en Estados Unidos a nivel federal,» 2017. [En línea]. Available: <http://www.gadgets.com/noticias/marco-legal-coche-autonomo-estados-unidos/>. [Último acceso:

- septiembre 2017].
- [37] A. Tributaria, «Tabla de coeficientes de amortización lineal,» 2017. [En línea]. Available: http://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos/_Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml. [Último acceso: septiembre 2017].
- [38] DERBLAUEMOND, «Éste es el mercado más importante que abre el coche autónomo,» 2017. [En línea]. Available: <https://www.elblogsalmon.com/sectores/este-es-el-mercado-mas-importante-que-abre-el-coche-autonomo-y-no-no-es-el-transporte>. [Último acceso: SEPTIEMBRE 2017].
- [39] P. J. Rousseeuw, «Least median of squares regression,» *Journal of the American Statistical Association*, 1984.
- [40] G. B. Villarreal, «Verificación de vehículos mediante técnicas de visión artificial,» 2014.